International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 03 | March - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

Data Security and Privacy Mechanism Based on Key Exposure Method over Cloud

Ajinkya Vinodrao Bhatkar

ABSTRACT

Recent news reveal a powerful attacker which breaks data confidentiality by acquiring cryptographic keys, by means of coercion or backdoors in cryptographic software. Once the encryption key is exposed, the only viable measure to preserve data confidentiality is to limit the attacker's access to the ciphertext. This may be achieved, for example, by spreading ciphertext blocks across servers in multiple administrative domains—thus assuming that the adversary cannot compromise all of them. Nevertheless, if data is encrypted with existing schemes, an adversary equipped with the encryption key, can still compromise a single server and decrypt the 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. 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.

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. 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 cipher text 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 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). 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 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. 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.

Objective of the Project

Recent news reveal a powerful attacker which breaks data confidentiality by acquiring cryptographic keys, by means of coercion or backdoors in cryptographic software. Once the encryption key is exposed, the only viable measure to preserve data confidentiality is to limit the attacker's access to the cipher text. This may be achieved, for example, by spreading cipher text blocks across servers in multiple administrative domains thus assuming that the adversary cannot compromise all of them. Nevertheless, if data is encrypted with existing schemes, an adversary equipped with the encryption key, can still compromise a single server and decrypt the cipher text 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 cipher text 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 cipher text 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.

2. LITERATURE SURVEY

2.1 RobuSTore: A Distributed Storage Architecture with Robust and High Performance

Emerging large-scale scientific applications require to access large data objects in high and robust performance. We propose RobuSTore, a storage architecture that combines erasure codes and speculative access mechanisms for parallel write and read in distributed environments. The mechanisms can effectively aggregate the bandwidth from a large number of distributed disks and statistically tolerate pear-disk performance variation. Our simulation results affirm the high and robust performance of RobuSTore in both write and read operations compared to traditional parallel storage systems. For example, for a 1GB data access using 64 disks, RobuSTore achieves average bandwidth of 186MBps for write and 400MBps for read, nearly 6x and 15x that achieved by a RAID-0 system. The standard deviation of access latency is only 0.5 second, about 9% of the write latency and 20% of the read latency, and a 5-fold improvement from RAID-0. The improvements are achieved at moderate cost: about 40% increase in I/O operations and 2x-3x increase in storage capacity utilization.We propose the distributed storage architecture RobuSTore for high and robust storage performance in distributed environments. Achieving high and robust performance in distributed storage systems is an important open research challenge. Traditional network file systems or local parallel file systems cannot satisfy these requirements. The performance variation of the individual disks is the major obstacle facing current systems. We propose the RobuSTore idea to address this issue. RobuSTore combines erasure coding and speculative access mechanisms for high and robust storage performance. The erasure coding mechanism encodes the original data into fragment blocks with symmetric redundancy, allowing flexible data striping during write accesses and flexible data reconstruction during read accesses. The speculative access mechanism fully utilizes the available disk bandwidths to read/write redundant fragment blocks from/to heterogeneous distributed disks. We then discussed the critical design choices for erasure coding and speculative access, which gave guidelines for the RobuSTore implementation. We compare the performance of RobuSTore with three traditional parallel storage schemes and see superior performance from RobuSTore in both write and read accesses. For example, for a 1GB data access using 64 disks with random in-disk data layout, RobuSTore achieves average bandwidth of 186MBps for write and 400MBps for read, nearly 6x and 15x that achieved by a RAID-0 system. The standard deviation of access latency is only 0.5 second, about 9% of the write latency and 20% of the read latency, and a 5-fold improvement from RAID-0. The improvements are achieved at moderate cost: about 40% increase in I/O operations and 2x-3x increase in storage capacity utilization.

We would remind the audience about the limitations of RobuSTore again. RobuSTore is not a general storage system; instead, it is for accessing large data objects on which update operations are rare. On homogeneous storage clusters with no shared access, RobuSTore is not better than traditional parallel file system due to the reception overhead of erasure

codes, which will reduce the bandwidth for up to 30%.

While we believe that we have made significant contributions, more advances can be made to improve the RobuSTore design and performance. First, we need erasure codes algorithms that can deliver higher coding bandwidth to match the network bandwidth increase. We achieve around 600 MBps decoding bandwidth with 50% reception overhead using LT Codes on 2.8 GHz AMD Opteron Processor. This is about 7 Gbps network utilization. Higher coding performance may be achieved by more efficient erasure codes, parallel coding algorithms, or dedicated coding hardware. Admission control is another topic that can complete the RobuSTore design. It is important for QoS guarantee and efficient resource sharing.

2.2 SPANStore: Cost-Effective Geo-Replicated Storage Spanning Multiple Cloud Services

By offering storage services in several geographically distributed data centres, cloud computing platforms enable applications to offer low latency access to user data. However, application developers are left to deal with the complexities associated with choosing the storage services at which any object is replicated and maintaining consistency across these replicas. In this paper, we present SPAN Store, a key-value store that exports a unified view of storage services in geographically distributed data centers. To minimize an application provider's cost, we combine three key principles. First, SPAN Store spans multiple cloud providers to increase the geographical density of data centers and to minimize cost by exploiting pricing discrepancies across providers. Second, by estimating application workload at the right granularity, SPAN Store judiciously trades off greater geo-distributed replication necessary to satisfy latency goals with the higher storage and data propagation costs this entails in order to satisfy fault tolerance and consistency requirements. Finally, SPAN Store minimizes the use of computer resources to implement tasks such as two-phase locking and data propagation, which are necessary to offer a global view of the storage services that it builds upon. Our evaluation of SPAN Store shows that it can lower costs by over 10x in several scenarios, in comparison with alternative solutions that either use a single storage provider or replicate every object to every data centre from which it is accessed.

Though the number of cloud storage services available across the world continues to increase, the onus is on application developers to replicate data across these services. We develop SPAN Store to export a unified view of geographically distributed storage services to applications and to automate the process of trading off cost and latency, while satisfying consistency and fault-tolerance requirements. Our design of SPAN Store achieves this goal by spanning data centres of multiple cloud providers, by judiciously determining replication policies based on workload properties, and by minimizing the use of computer resources. We have deployed SPAN Store across Amazon's, Microsoft's, and Google's cloud services and find that it can offer significant cost benefits compared to simple replication policies.

2.3 Evolution of Cloud Storage as Cloud Computing Infrastructure Service

Enterprises are driving towards less cost, more availability, agility, managed risk - all of which is accelerated towards Cloud Computing. Cloud is not a particular product, but a way of delivering IT services that are consumable on demand, elastic to scale up and down as needed, and follow a pay-for-usage model. Out of the three common types of cloud computing service models, Infrastructure as a Service (IaaS) is a service model that provides servers, computing power, network bandwidth and Storage capacity, as a service to their subscribers. Cloud can relate to many things but without the fundamental storage pieces, which is provided as a service namely Cloud Storage, none of the other applications is possible. This paper introduces Cloud Storage, which covers the key technologies in cloud computing and Cloud Storage, management insights about cloud computing, different types of cloud services, driving forces of cloud computing and cloud storage, advantages and challenges of cloud storage and concludes by pinpointing few challenges to be addressed by the cloud storage providers.

Cloud computing represents the next evolutionary step toward elastic IT. Cloud emergence transforms the way in which IT infrastructure is constituted and managed through consumable services for infrastructure, platform, and applications. This idea converts IT infrastructure from a “factory” into a “supply chain”. There may be a stage to come when Internet is going to be the communication channel for mass media, then we cannot imagine a world without cloud storage because keeping ownership and maintaining huge volume of data on our own infrastructure is unimaginable. So automatically cloud storage will capture the entire market as we see rental houses leased to the tenants which is an unavoidable and a must situation in a growing populated city. Cloud storage strategies and service models are still in its early stages. Standardization of service provider’s service levels, pricing plans, data access methods, operational and security processes, emergency plans for data migration if the enterprise sooner or later wish to change vendors, improving the performance by opting better load balancing methodology are some of the thrust areas where future works on cloud storage can be focused.

3.4 AONT-RS: Blending Security and Performance in Dispersed Storage Systems

Dispersing files across multiple sites yields a variety of obvious benefits, such as availability, proximity and reliability. Less obviously, it enables security to be achieved without relying on encryption keys. Standard approaches to dispersal either achieve very high security with correspondingly high computational and storage costs, or low security with lower costs. In this paper, we describe a new dispersal scheme, called AONT-RS, which blends an All-Or-Nothing Transform with Reed-Solomon coding to achieve high security with low computational and storage costs. We evaluate this scheme both theoretically and as implemented with standard open source tools. AONTRS forms the backbone of a commercial dispersed storage system, which we briefly describe and then use as a further experimental testbed. We conclude with details of actual deployments.

Dispersed storage systems enable availability, scalability, and performance based on physical proximity. They also enable security via (k,n) threshold schemes that require attackers to authenticate themselves to k of n storage nodes in order to read data. The threshold schemes provide this security without relying on the secure storage of encryption keys, which is a notoriously difficult problem.

We have described a new dispersal algorithm called AONT-RS, which combines the All-Or-Nothing Transform with systematic Reed-Solomon codes to achieve computational security. Compared to traditional approaches to dispersal, AONT-RS has a very attractive blend of properties. Its storage and computational footprint is much less than Shamir secret sharing. While Shamir achieves information theoretic security AONTRS's security can be tuned so that compromise is computationally infeasible. Compared to Rabin's classic dispersal algorithm, AONT-RS achieves a far greater degree of security, and also better performance for larger installations. This is because AONT-RS is based on a systematic Reed-Solomon erasure code rather than the non-systematic code employed by Rabin. We have detailed the theoretical and applied performance of the dispersal algorithms, and described a commercial dispersed storage product that is based upon the dispersal algorithm. AONT-RS is not specific to our dispersal solution. For example, the POTSHARDS archival storage system could use AONT-RS to implement computational rather than information theoretic security and reduce their storage requirements by a factor of three. Other solutions such as Grid sharing can improve their security byemploying AONT-RS rather than a standard systematic Reed-Solomon code.

OceanStore: Architecture for Global-Scale Persistent Storage

Ocean Store is a utility infrastructure designed to span the globe and provide continuous access to persistent information. Since this infrastructure is comprised of untrusted servers, data is protected through redundancy and cryptographic techniques. To improve performance, data is allowed to be cached anywhere, anytime. Additionally, monitoring of usage patterns allows adaptation to regional outages and denial of service attacks; monitoring also enhances performance through pro-active movement of data. A prototype implementation is currently under development.The rise of ubiquitous computing has spawned an urgent need for persistent information. In this paper we presented OceanStore, a utility infrastructure designed to span the globe and provide secure, highly available access to persistent objects. Several properties distinguish OceanStore from other systems: the utility model, the untrusted infrastructure, support for truly nomadic data, and use of introspection to enhance performance and maintainability. A utility model makes the notion of a global system possible, but introduces the possibility of untrustworthy servers in the system. To this end, we assume that servers may be run by adversaries and cannot be trusted with cleartext; as a result, serverside operations such as conflict-resolution must be performed directly on encrypted information. Nomadic data permits a wide range of optimizations for access to information by bringing it "close" to where it is needed, and enables rapid response to

regional outages and denial-of-service attacks. These optimizations are assisted by introspection, the continuous online collection and analysis of access patterns. OceanStore is under construction. This paper presented many of the design elements and algorithms of OceanStore; several have been implemented. Hopefully, we have convinced the reader that an infrastructure such as OceanStore is possible to construct; that it is desirable should be obvious.

A Receiver Deniable Encryption Scheme

A practical efficient receiver deniable encryption scheme based on BCP commitment scheme and idea of Klonowski et al. is proposed in the paper. The analysis of the proposed schemes is also presented.

In this paper we propose a practical efficient receiver deniable encryption scheme based on BCP commitment scheme and idea of Klonowski et al. The proposed scheme is a one-move scheme without any pre-encryption information required to be sent between the sender and the receiver prior to encryption. Moreover, the overhead is low in term of the size of the ciphertext.

Deniable Encryption Protocols Based on Probabilistic Public-Key Encryption

The paper proposes a new method for designing deniable encryption protocols characterized in using RSA-like probabilistic public-key encryption algorithms. Sender-, receiver-, and bi-deniable protocols are described. To provide bi-deniability in the case of attacks performed by an active coercer stage of entity authentication is used in one of described protocols.

There have been proposed flexible sender-side, receiverside, and sender&receiver-side DE protocols that are very attractive from practical point of view due to their providing super-polynomial resistance to coercive attacks and comparatively high performance. The proposed design can be potentially extended on the case of combining the commutative functions with probabilistic public-key encryption based on computational difficulty of the discrete logarithm problem on an elliptic curve. This case will give potentially exponential resistance to coercive attack. Besides, this research direction can give DE protocols more suitable for using standard public key infrastructure, like plan-ahead public-key DE protocol introduced in paper.

It has been also proposed fully bi-deniable public encryption protocol with plan-ahead fake message, which is secure against active attacks. The last DE scheme includes steps of the entity mutual authentication in frame of which the parties of the protocol they hide execution of the procedureof exchanging the single-use public keys. The lasts are used to mask the ciphertext containing the secrete message.

HYDRAstor: a Scalable Secondary Storage

HYDRAstor is a scalable, secondary storage solution aimed at the enterprise market. The system consists of a back-end architecture as a grid of storage nodes built around a distributed hash table; and a front-end consisting of a layer of access nodes which implement a traditional file system interface and can be scaled in number for increased performance. This paper concentrates on the back-end which is, to our knowledge, the first commercial implementation of a

scalable, high-performance content-addressable secondary storage delivering global duplicate elimination, per-block user-selectable failure resiliency, self maintenance including automatic recovery from failures with data and network overlay rebuilding.

The back-end programming model is based on an abstraction of a sea of variable-sized, content-addressed, immutable, highly-resilient data blocks organized in a DAG (directed acyclic graph). This model is exported with low-level API allowing clients to implement new access protocols and to add them to the system on-line. The API has been validated with an implementation of the file system interface. The critical factor for meeting the design targets has been the selection of proper data organization based on redundant chains of data containers. We present this organization in detail and describe how it is used to deliver required data services. Surprisingly, the most complex to deliver turned out to be on-demand data deletion, followed (not surprisingly) by the management of data consistency and integrity.

HYDRAstor is a decentralized, scalable secondary storage that is commercially available today. It can be used as an on-line repository for all enterprise backup and archival data while dynamically and efficiently sharing available capacity. Critical features like high-availability and reliability, ease of management, capacity and performance scalability, and storage efficiency make the system unique in addressing today's enterprise needs. The system is externally visible as one storage pool and can be accessed by legacy applications using traditional file system interface.

The core architecture is built around a DHT with virtual super nodes spanned over physical nodes. Data resiliency is provided with erasure codes, with fragments of erasure-coded blocks distributed among super node components. Redundancy in the network and data allows for on-line upgrades and extensions, increasing availability of the system. High storage efficiency is facilitated by variable block size global deduplication. The back-end exports low-level API providing operations on content addressed blocks which expose pointers to other blocks. A novel data organization based on redundant chains of data containers is used to deliver reliably multitude of data services, including failure-tolerant deletion and fast verification of data health.

Although the system is fully functional today, there is an important work left to improve its value delivered to the end user. The read-only phase of deletion will be eliminated, which will make the system fully usable all the time. Deduplication can be moved to a proxy server, saving bandwidth and improving write performance of highly-duplicated streams. Additionally, since multiple types of drivers can write to the back-end, there is a need for a stream interface that can cut data into blocks in a standard way. This will ensure higher deduplication among data written by different types of clients.

The Security of All-Or-Nothing Encryption: Protecting Against Exhaustive Key Search

We investigate the all-or-nothing encryption paradigm which was introduced by Rivest as a new

mode of operation for block ciphers. The paradigm involves composing an all-or-nothing transform (AONT) with an ordinary encryption mode. The goal is to have secure encryption modes with the additional property that exhaustive key-search attacks on them are slowed down by a factor equal to the number of blocks in the ciphertext.

We give a new notion concerned with the privacy of keys that provably captures this key-search resistance property. We suggest a new characterization of AONTs and establish that the resulting all-or-nothing encryption paradigm yields secure encryption modes that also meet this notion of key privacy. A consequence of our new characterization is that we get more efficient ways of instantiating the all-or-nothing encryption paradigm. We describe a simple block-cipher-based AONT and prove it secure in the Shannon Model of a block cipher. We also give attacks against alternate paradigms that were believed to have the above key search resistance property.

Broadcast Enforced Threshold Schemes with Disenrollment

Blakley, Blakley, Chan and Massey conjectured a lower bound on the entropy of broadcast messages in threshold schemes with disenrollment. In an effort to examine the conjecture, we identify their original scheme definition has a limitation: a coalition of participants can reconstruct all shared secrets without broadcast from the dealer, and hence render the dealer no control over disenrollment. We introduce a constraint that delays this lack of control of the dealer over disenrollment. We also establish the lower bounds on the entropy of broadcast messages in such a model. We demonstrate the need for new models by presenting a construction under open problems.

During the process of examining the conjecture on the lower bound of broadcast entropy, we found that the original model of threshold schemes with disenrollment is inadequate to ensure the dealer of the control over each disenrollment. We presented a broadcast enforced model which ensures that the public broadcast from the dealer is required for disenrollment, by adding an additional term to the original definition. We showed that in related previous work to establish a lower bound on broadcast size, though the original model is used, the validity of LEMMA 2 does require the broadcast from the dealer. In the new model, the coalition of any number of participants is unable to reconstruct the shared secret Ki before the ith disenrollment stage. We also derived lower bounds on the entropy of broadcast messages in such a model, which are refined from the bound obtained in. There is an open problem with threshold scheme with disenrollment.

An Unconditional Cheating Detection and Cheater Identification in Shamir's Secret Sharing Scheme

Secret sharing scheme consist to divide a secret Danto n parts in order to easily recover it from any number of k parts, but k-1 parts reveal no information about D. During reconstruction phase, some of the shareholders (dishonest) may cheat by showing fake shared key. As a result, the remaining shareholders (honest) cannot get true secret. A good secret sharing scheme must detect and identify those shareholders called cheaters. In this paper, we improve upon the Shamir secret sharing scheme by incorporating

cheater detection and identification capability. This paper uses one way hash function and arithmetic concept. The proposed algorithm is unconditional and secure and will be able to detect and identify any cheater.

In this paper, we proposed a solution for cheating detection and cheater identification. The secret sharing scheme used is Shamir's one. The detection and identification is performed by using one way hash function and arithmetic concept. Our solution is unconditional and secure. However, we denoted the fact that, it is lacking the capability to avoid cheater recovering the legal secret when cheating is detected.

3. ANALYSIS

Introduction

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of software development methodologies. These methodologies form the framework for planning and controlling the creation of an information system the software development process.

EXISTING SYSTEM

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. 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.

Disadvantages of Existing System:

1. 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).

PROPOSED SYSTEM

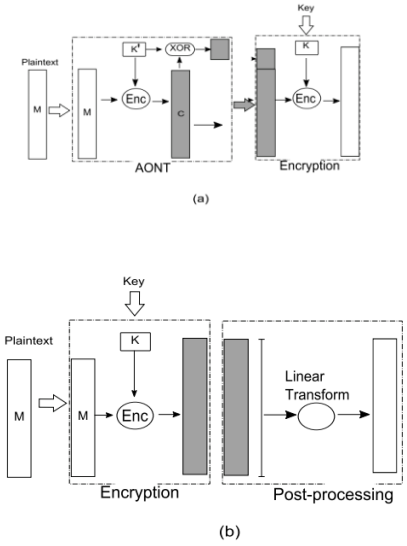
To counter such existing adversaries we propose Bastion, a novel and efficient scheme which ensures that plaintext data cannot be recovered as long as the adversary has access to at most all but two ciphertext blocks, even when the encryption key is exposed. We analyze the security of Bastion, and we show that it prevents leakage of any plaintext block as long as the adversary has access to the encryption key and to all but two ciphertext blocks. We evaluate the performance of Bastion analytically and empirically

in comparison to a number of existing encryption techniques.

Advantages of Proposed System:

1. 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

SYSTEM IMPLEMENTATION



Modules

1. Post-Processing module
2. Encryption Module

Post-Processing

Bastion first encrypts the data with one round of block cipher encryption, and then applies an efficient linear post-processing to the ciphertext. By doing so, Bastion relaxes the notion of all-or-nothing encryption at the benefit of increased performance.

Encryption

More specifically, the first round of Bastion consists of CTR mode encryption with a randomly chosen key. The output ciphertext y′ is then fed to a linear transform.

SYSTEM REQUIREMENTS Hardware Requirements:

- Processor - Pentium –IV
- Speed - 1.1 GHz
- Ram - 256 MB
- Hard Disk - 20 GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA

Software Requirements:

- Operating System -Windows XP
- Coding Language - Java

Chapter 5

Conclusion and Future Scope

In this project, we addressed the problem of securing data outsourced to the cloud against an adversary which has access to the encryption key. For that purpose, we introduced a novel security definition that captures data confidentiality against the new adversary. We then proposed Bastion, a scheme 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. We analyzed the security of Bastion 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.

Chapter 6 References

[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. 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. [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.