A NOVEL APPROACH FOR DATA CONFIDENTIALITY UNDER KEY EXPOSURE IN CLOUDS

R.Sivaiah, siva.r140@gmail.com

Assistant Professor, Department of Computer Science Engineering , Narayana Engineering College, Gudur, AP, India 524101

M.Venkata Nikitha, V.Sravani, P.Priyadharshini, N.Keerthana

nikitha.mv123@gmail.com , sravani05b0@gmail.com , puchalapallipriyadarshini@gmail.com, ,keerthanaraj.n62@gmail.com

UG Student, Department of Computer Science Engineering , Narayana Engineering College, Gudur, AP, India 524101

Abstract – Recent surveys reveals 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. 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[8].

Index terms – Data Confidentiality, Key exposure, dispersed storage.

I. 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 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, 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 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, as long as 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 contributions in this paper can be summarized as follows:

- We propose Bastion, an efficient scheme which ensures data confidentiality against an adversary that knows the encryption key and has access to a large fraction of the ciphertext blocks.
- 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. Our 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).

We discuss practical insights with respect to the deployment of Bastion within existing storage systems, such as the HYDRA stor grid storage system.

II. BACKGROUND WORK

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.

A) *Deniable Encryption*

Our work shares similarities with the notion of "shared key deniable encryption". 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.

B) *Information Dispersal*

Information dispersal based on erasure codes has been proven 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 servers failures.

Ramp schemes [7] 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 and features 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.

C) *All or Nothing Transformations*

All-or-nothing transformations (AONTs) were first introduced in and later studied. 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. combine AONT and information dispersal to provide both fault-tolerance and data secrecy, in the context of distributed storage systems. In [9], however, an adversary which knows the encryption key can decrypt data stored on single servers.

D) *Secret Sharing*

Secret sharing schemes [5] 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 [10] 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, however the proposal does not provide any security guarantees when a small number of shares (less than the reconstruction threshold) 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, while the encryption key is shared using the scheme. In Krawczyk's scheme, individual ciphertext blocks encrypted with AES can be decrypted once the key is exposed.

III. PROPOSED WORK

In this section, we present our scheme, dubbed Bastion, which ensures that plaintext data cannot be recovered as long as the adversary has access to all but two ciphertext blocks—even when the encryption key is exposed.

Bastion departs from existing AON encryption schemes. Current schemes require a pre-processing round of block cipher encryption for the AONT, followed by another round of block cipher encryption (cf. Figure 1 (a)). Differently, Bastion first encrypts the data with one round of block cipher encryption, and then applies an efficient linear post-processing to the ciphertext (cf. Figure 1 (b)). By doing so, Bastion relaxes the notion of all-or-nothing encryption at the benefit of increased performance (see Figure 1).

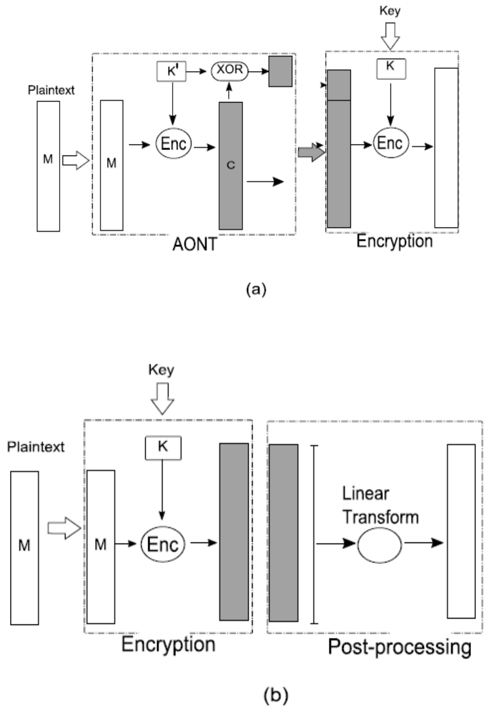
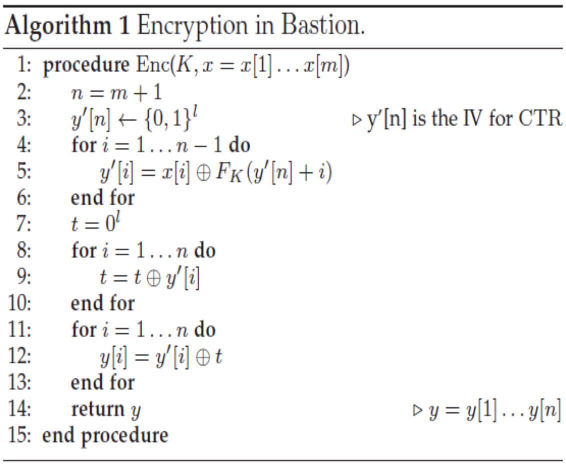


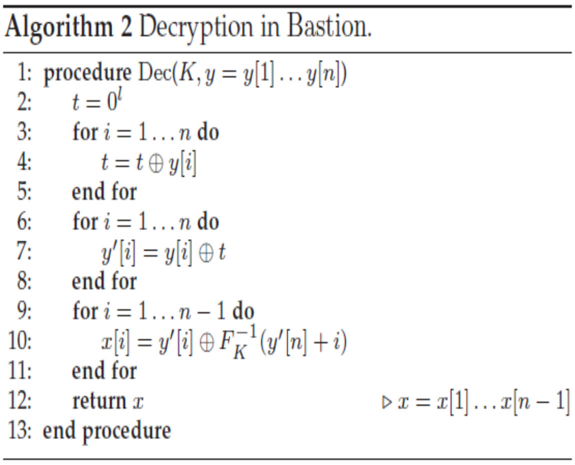
Fig. 1: (a) Current AON encryption schemes require a pre-processing round of block cipher encryption for the AONT, followed by another round of block cipher encryption. (b) On the other hand, Bastion first encrypts the data with one round of block cipher encryption, and then applies an efficient linear post-processing to the ciphertext.

The encryption and decryption of the Bastion are shown in the following Algorithm 1 and Algorithm 2.



Algorithm 1 Encryption in Bastion.

```
procedure Enc($K, x = x[1] \ldots x[m]$)
    $n = m + 1$
    $y'[n] \leftarrow \{0,1\}^l$                    ▷ y'[n] is the IV for CTR
    for $i = 1 \ldots n - 1$ do
        $y'[i] = x[i] \oplus F_K(y'[n] + i)$
    end for
    $t = 0^l$
    for $i = 1 \ldots n$ do
        $t = t \oplus y'[i]$
    end for
    for $i = 1 \ldots n$ do
        $y[i] = y'[i] \oplus t$
    end for
    return $y$                                      ▷ $y = y[1] \ldots y[n]$
end procedure
```



Algorithm 2 Decryption in Bastion.

```
1: procedure Dec($K, y = y[1] \dots y[n]$)
2:     $t = 0^l$
3:     for $i = 1 \dots n$ do
4:         $t = t \oplus y[i]$
5:     end for
6:     for $i = 1 \dots n$ do
7:         $y'[i] = y[i] \oplus t$
8:     end for
9:     for $i = 1 \dots n - 1$ do
10:        $x[i] = y'[i] \oplus F_K^{-1}(y'[n] + i)$
11:    end for
12:    return $x$                    ▷ $x = x[1] \dots x[n-1]$
13: end procedure
```

Therefore, we are only left to show that the linear transformation computed in lines 7-14 of Algorithm 1 is correctly reverted in lines 2-8 of Algorithm 2.

IV. SYSTEM IMPLEMENTATION

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.

V. CONCLUSION

In this paper, 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. Finally, we showed how Bastion can be practically integrated within existing dispersed storage systems.

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.
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. Mandava Geetha Bhargava, Modugula TS Srinivasa Reddy, Shaik Shahbaz, P Venkateswara Rao, V Sucharita Potential of big data analytics in bio-medical and health care arena: An exploratory study, Global Journal of Computer Science and Technology 2017/8/5
9. U.Ambili, P.Rajasekar, (2017), "Area and Power Optimized Lightweight PRESENT-GRP Cryptography Algorithm",International Journal of Emerging Innovations in Science and Technology(ISSN:2348-439X) Volume 3, issue4.
10. V.Sucharita, P.Ravinder Rao,"A Framework to Automate Cloud based Service Attacks Detection and Prevention"(IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 10, No. 2, 2019
11. Kollu, A., Sucharita, V. (2018). Energy-Aware Multi-objective Differential Evolution in Cloud Computing. In: Dash, S., Das, S., Panigrahi, B. (eds) International Conference on Intelligent Computing and Applications. Advances in Intelligent Systems and Computing, vol 632. Springer,Singapore. https://doi.org/10.1007/978-981-10-5520-1_40