International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 4 | Issue 2 | ISSN : 2456-3307

Efficient Key Exposure Method Implementation In Cloud Storage

[1*]K. Sandhya Rani, Kommalapati Rajesh[2]

ABSTRACT

Late news uncovers a capable assailant, which breaks information classification by gaining cryptographic keys, by methods for intimidation or indirect accesses in cryptographic programming. Once the encryption key is uncovered, the main practical measure to safeguard information secrecy is to restrain the assailant's entrance to the cipher text. This might be accomplished, for instance, by spreading cipher text blocks crosswise over servers in different authoritative areas—along these lines expecting that the enemy cannot trade off every one of them. Overall, if information is encoded with existing plans, a foe outfitted with the encryption key, can in any case bargain a solitary server and decode the cipher text blocks put away in that. In this paper, we consider information privacy against a foe, which knows the encryption key and approaches a huge division of the cipher text blocks. To this end, we propose BASTION, a novel and effective plan that ensures information classification regardless of whether the encryption key is spilled and the enemy approaches all cipher text blocks. We break down the security of BASTION, and we assess its execution by methods for a model usage. We likewise talk about down to earth bits of knowledge concerning the mix of BASTION in business scattered capacity frameworks. Our assessment comes about recommend that BASTION is appropriate for coordination in existing frameworks since it brings about under 5% overhead contrasted with existing semantically secure encryption modes.

Keywords: Conjointly, Traversal, Exposure, Auditing, Investigate

I. INTRODUCTION

Cloud computing could be a computing paradigm, wherever an oversized pool of systems square measure connected privately or public networks, to produce dynamically ascendable infrastructure for application, information and file storage. The big quantity of knowledge is hold on within the cloud. To verify the integrity of information that is hold on the cloud, the cloud storage auditing is employed. Auditing is associate integrity sign in the cloud information base. It is a very important checking within the cloud auditing protocols that square measure extremely researched on recent years. Every protocols act as a unique auditing mechanism. The aim of introducing the protocol is to realize high information measure and computation potency.

THE world recently witnessed a massive surveylance program aimed at breaking users' privacy. Perpetrators were not hindered by the various security measures deployed within the targeted services [31]. 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 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 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 [31], 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 that 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 en-crypt ion functions with an efficient linear transform. In this sense, Bastion shares similarities with the no tin 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 ac-cuss 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 pre-processing round to create the AONT, followed by an-other 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 compare-son with a number of existing encryption schemes. Our results show that Bastion only incurs a negligible per- romance deterioration (less than 5%) when compared to symmetric encryption schemes, and considerably improves the performance of existing AON encryption schemes [12], [26]. We also discuss practical insights with respect to the possible integration of Bastion in commercial dispersed storage systems.

II. RELATED 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.

Deniable Encryption

Our work shares similarities with the notion of “shared-key deniable encryption” [9], [14], [18]. An encryption scheme is “deniable” if—when coerced to reveal the encryption key—the legitimate owner reveals “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 en-crypt ion 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 Dispersal

Information dispersal based on erasure codes [30] has been proven as an effective tool to provide reliability in a number of cloud-based storage systems [1], [2], [20], [33]. Erasure codes enable users to distribute their data on a number of servers and recover it despite some server’s 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 t_1, t_2. At least t_2 shares are required to reconstruct the secret and less than t_1 shares provide no information about the secret; a number of shares between t_1 and t_2 leak "some" information.

All or Nothing Transformations

All-or-nothing transformations (AONTs) were first introduced in [26] and later studied in [8], [12]. 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. Reach et al. [25] combine AONT and information dispersal to provide both fault-tolerance and data secrecy, in the context of distributed storage systems. In [25], however, an adversary, which knows the encryption key, can decrypt data stored on single servers.

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 [11], [27], 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 [24] proposed an information dispersal algorithm with smaller overhead than the one of [27], however the proposal in [24] 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 [27] and Ra-bin's [24] approaches; in [19] a file is first encrypted using AES and then dispersed using the scheme in [24], while the encryption key is shared using the scheme in [27]. 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 cryp-tographic primitives that can resist an adversary which learns partial information about the secret state of a sys-tem, e.g., through side-channels [22]. Different models allow to reason about the "leaks" of real implemen-tations of cryptographic primitives [22]. 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.

Existing system

III. SYSTEM MODEL

We consider a multi-cloud storage system which can leverage a number of commodity cloud providers (e.g., Amazon, Google) with the goal of distributing trust across different administrative domains. This "cloud of clouds" model is receiving increasing attention nowa-days [4], [6], [32] with cloud storage providers such as EMC, IBM, and Microsoft, offering products for multi-cloud systems [15], [16], [29].

In particular, we consider a system of s storage servers S1, . . . , Ss, and a collection of users. We assume that each server appropriately authenticates users. For simplicity and without loss of generality, we focus on the read/write storage abstraction of [21] which exports two operations:

WRITE(v) This routine splits v into s pieces {v1, . . . , vs} and sends hvj i to server Sj , for j ∈ [1 . . . s].

READ(•) The read routine fetches the stored value v from the servers. For each j ∈ [1 . . . s], piece vj is downloaded from server Sj and all

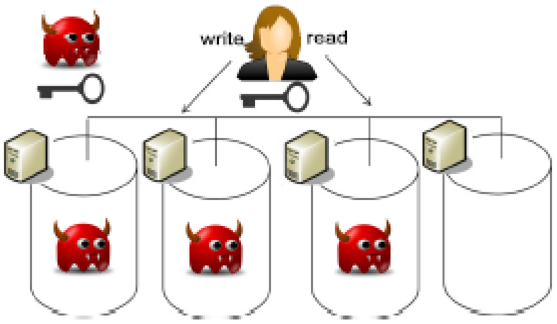


Figure 1. Our attacker model. We assume an adversary which can acquire all the cryptographic secret material, and can compromise a large fraction (up to all but one) of the storage servers.

3.2 Adversarial Model

We assume a computationally-bounded adversary A which can acquire the long-term cryptographic keys used to encrypt the data. The adversary may do so either (i) by leveraging flaws or backdoors in the key-generation software [31], or (ii) by compromising the device that stores the keys (in the cloud or at the user). Since cipher text blocks are distributed across servers hosted within different domains, we assume that the adversary cannot compromise all storage servers (cf. Figure 1). In particular, we assume that the adversary can compromise all but one of the servers and we model this adversary by giving it access to all but λ cipher text blocks.

Note that if the adversary also learns the user's credentials to log into the storage servers and downloads all the cipher text blocks, then no cryptographic mech-anism can preserve data confidentiality. We stress that compromising the encryption key does not necessarily imply the compromise of the user's credentials. For example, encryption can occur on a specific-purpose device [10], and the key can be leaked, e.g., by the manufacturer; in this scenario, the user's credentials to access the cloud servers are clearly not compromised.

3.3 (n – λ)-CAKE Security

Existing security notions for encryption modes capture data confidentiality against an adversary which does not have the encryption key. That is, if the key is leaked, the confidentiality of data is broken.

In this paper we study an adversary that has access to the encryption key but does not have the entire ciphertext. We therefore propose a new security definition that models our scenario.

As introduced above, we allow the adversary to access an encryption/decryption oracle and to "see" all but λ cipher text blocks. Since confidentiality with λ = 0 is clearly not achievable1, we instead seek an encryption mode where λ = 1. However, having the flexibility of setting λ ≥ 1 allows the design of more efficient schemes while keeping a high degree of security in practical deployments. (See Remark 7.)

We call our security notion (n–λ) Ciphertext Access under Key Exposure, or (n – λ)CAKE. Similar to [12], (n – λ)CAKE specifies a block length l such that a cipher text y can be written as y = y[1] . . . y[n] where |y[i]| = l and n > 1.

```
Exp(n–λ)CAKE(A, b)
a          ← K(1k)
EFA ,F –1
x0, x1, state ← A          A          (f ind)
yb         ← EFA ,FA–1 (xb)
'          Y ,EFA ,F –1
b ← A B          A          (guess, state)
```

The adversary has unrestricted access to EFA ,FA–1 in both the "find" and "guess" stages. On input j, the oracle Yb returns yb[j] and accepts up to n – λ queries. On the one hand, unrestricted oracle access to EFA ,FA–1 captures the adversary's knowledge of the secret key. On the other hand, the oracle Yb models the fact that the adversary has access to all

but λ cipher text blocks. This is the case when, for example, each server stores λ cipher text blocks and the adversary cannot compromise all servers. The advantage of the adversary is defined as:
Adv(n–λ)CAKE(A) = P r[Exp(n–λ)CAKE(A, 1) = 1]–
P r[Exp(n–λ)CAKE(A, 0) = 1]
DEfiNITION 3. An encryption mode Q = (K, E, D) is (n – λ)CAKE secure if for any p.p.t. adversary A, we have Adv(n–λ)CAKE(A) ≤ ϙ, where ϙ is a negligible function in the security parameter.

Definition 3 resembles Definition 2 but has two fundamental differences. First, (n – λ)CAKE refers to a keyed scheme and gives the adversary unrestricted access to the encryption/decryption oracles. Second, (n – λ)CAKE relaxes the notion of all-or-nothing and parameterizes the number of ciphertext blocks that are not given to the adversary. As we will show in Sec-tion 4.2, this relaxation allows us to design encryption modes that are considerably more efficient than existing modes which offer a comparable level of security.

We stress that (n – λ)CAKE does not consider confidentiality against "traditional" adversaries (i.e., adver-saries which do not know the encryption key). Indeed, an ind-adversary is not given the encryption key but has access to all ciphertext blocks. That is, the ind-adversary can compromise all the s storage servers. An (n – λ)CAKE-adversary is given the encryption key but can access all but λ ciphertext blocks. In practice, the (n – λ)CAKE-adversary has the encryption key but can compromise up to s – 1 storage servers. Therefore,we seek an encryption mode with the following properties:

1) must be ind secure against an adversary which does not know the encryption key but has access to all ciphertext blocks (cf. Definition 1), by compromising all storage servers.

2) must be (n – λ)CAKE secure against an adversary which knows the encryption key but has access to n – λ ciphertext blocks (cf. Definition 3), since it cannot compromise all storage servers.

REMARK 4. Property 2 ensures data confidentiality against the attacker model outlined in Section 3.2. Nevertheless, we must also account for weaker adversaries (i.e., traditional adversaries) that do not know the encryption key but can access the entire ciphertext —hence, ind security. Note that if the adversary which has access to the encryption key, can also access all the ciphertext blocks, then no cryptographic mechanism can preserve data confidentiality.

IV. PROPOSE METHODS

BASTION: SECURITY AGAINST KEY EXPO-SURE

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. We then analyze the security of Bastion with respect to Definition 1 and Definition 3.

4.1 Overview

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 2 (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 2 (b)). By doing so, Bastion relaxes the notion of all-or-nothing encryption at the benefit of increased performance (see Figure 2).

More specifically, the first round of Bastion consists of CTR mode encryption with a randomly chosen key K, i.e., $y' = Enc(K, x)$. The output ciphertext y' is then fed to a linear transform which is inspired by the scheme of [28]. Namely, our transform basically com-putes $y = y' \cdot A$ where A is a square matrix such that: *(i)* all diagonal elements are set to 0, and *(ii)* the remaining off-diagonal elements are set to 1. As we shown later, such a matrix is invertible and has the nice property that $A^{-1} = A$. Moreover, $y = y' \cdot A$ ensures that each input block y_j' will depend on all output blocks y_i except from y_j . This transformation—combined with the fact that the original input blocks have high entropy (due to semantic secure encryption)—result in an ind-secure and (n − 2)*CAKE* secure encryption mode. In the following section, we show how to efficiently compute $y' \cdot A$ by means of bitwise XOR operations.

4.2 Bastion: Protocol Specification We now detail the specification of Bastion.

On input a security parameter k, the key generation algorithm of Bastion outputs a key $K \in \{0, 1\}^k$ for the underlying block-cipher. Bastion leverages block cipher encryption in the CTR mode, which on input a plaintext bitstream x, divides it in blocks x[1], . . . , x[m], where m is odd[2] such that each block has size l.[3] The set of input blocks is encrypted under key K, resulting in ciphertext $y' = y'[1], \ldots, y'[m + 1]$, where $y'[m + 1]$ is an initialization vector which is randomly chosen from

$\{0, 1\}^l$.

Next, Bastion applies a linear transform to y′ as follows. Let n = m + 1 and assume A to be an n-by-n matrix where element $a_{i,j} = 0^l$ if $i = j$ or $= 1^l$, otherwise.[4] Bastion computes $y = y' \cdot A$,where additions and multiplications are implemented by means of XOR and AND operations, respectively.

That is, $y[i] \in y$ is computed as $y[i] = {}^{Lj=n}(y'[j] \wedge a_{j,i})$,

j=1

for i = 1 . . . , n.

Given key K, inverting Bastion entails computing $y' = y \cdot A^{-1}$ and decrypting y′ using K. Notice that matrix A is invertible and $A = A^{-1}$. The pseudocode of the encryption and decryption algorithms of Bastion are shown in Algorithms 1 and 2, respectively. Both algorithms use F to denote a generic block cipher (e.g., AES).

In our implementation, we efficiently compute the linear transform using 2n XOR operations as follows:

$t = y'[1] \oplus y'[2] \oplus \cdots \oplus y'[n]$, $y[i] = t \oplus y'[i]$, $1 \leq i \leq n$.

Note that $y'[1] \ldots y'[n]$ (computed up to line 6 in Algo-rithm 1) are the outputs of the CTR encryption mode, where $y'[n]$ is the initialization vector. Similar to the CTR encryption mode, the final output of Bastion is one block larger than the original input.

4.3 Correctness Analysis

We show that for every $x \in \{0, 1\}^{lm}$ where m is odd, and for every $K \in \{0, 1\}^l$, we have x =

Dec(K, Enc(K, x)).

In particular, notice that lines 2-6 of Algorithm 1 and lines 9-12 of Algorithm 2 correspond to the standard CTR encryption and decryption routines, respectively.

This requirement is essential for the correctness of the sub-sequent linear transform on the ciphertext blocks. That is, if m is even, then the transform is not invertible.

l is the block size of the particular block cipher used.

0^l and 1^l denote a bitstring of l zeros and a bitstream of l ones, respectively.

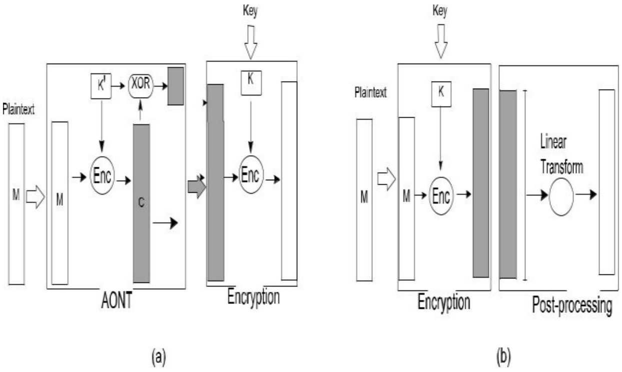
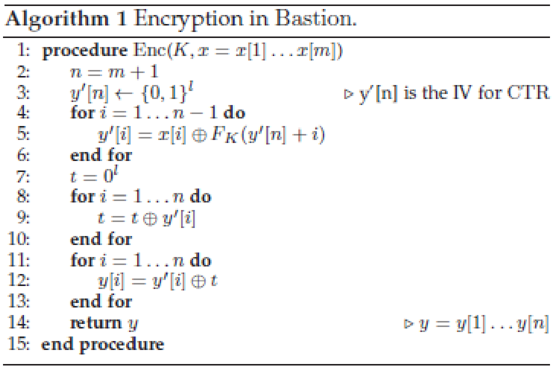


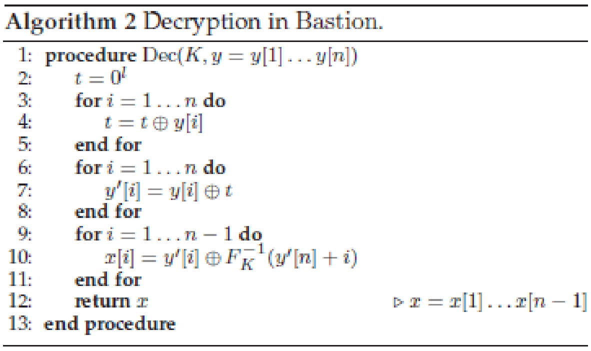
Figure 2. (a) Current AON encryption schemes require a pre-processing round of block cipher encryption for the AONT, followed by another round of block cipher encryption. (b) On the other hand, BASTION first encrypts the data with one

round of block cipher encryption, and then applies an efficient linear post-processing to the ciphertext.



Algorithm 1 Encryption in Bastion.

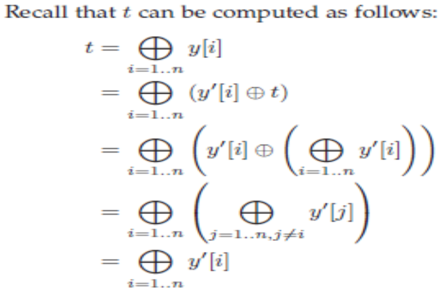
```
1:  procedure Enc($K, x = x[1] \ldots x[m]$)
2:      $n = m + 1$
3:      $y'[n] \leftarrow \{0,1\}^l$                    ▷ y'[n] is the IV for CTR
4:      for $i = 1 \ldots n-1$ do
5:          $y'[i] = x[i] \oplus F_K(y'[n] + i)$
6:      end for
7:      $t = 0^l$
8:      for $i = 1 \ldots n$ do
9:          $t = t \oplus y'[i]$
10:     end for
11:     for $i = 1 \ldots n$ do
12:         $y[i] = y'[i] \oplus t$
13:     end for
14:     return $y$                                ▷ $y = y[1] \ldots y[n]$
15: end procedure
```



Algorithm 2 Decryption in Bastion.

```
1:  procedure Dec($K, y = y[1] \ldots y[n]$)
2:      $t = 0^l$
3:      for $i = 1 \ldots n$ do
4:          $t = t \oplus y[i]$
5:      end for
6:      for $i = 1 \ldots n$ do
7:          $y'[i] = y[i] \oplus t$
8:      end for
9:      for $i = 1 \ldots n-1$ do
10:         $x[i] = y'[i] \oplus F_K^{-1}(y'[n] + i)$
11:     end for
12:     return $x$                                ▷ $x = x[1] \ldots x[n-1]$
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. In other words, we need to show that t = ^{L} y[i] (as computed in the decryption algorithm) matches = $^{L}{}_{i=1..n}$ y'[i] (as computed in the encryption algo-rithm).



Recall that t can be computed as follows:

$$\begin{aligned} t &= \bigoplus_{i=1..n} y[i] \\ &= \bigoplus_{i=1..n} (y'[i] \oplus t) \\ &= \bigoplus_{i=1..n} \left(y'[i] \oplus \left(\bigoplus_{i=1..n} y'[i] \right) \right) \\ &= \bigoplus_{i=1..n} \left(\bigoplus_{j=1..n, j \neq i} y'[j] \right) \\ &= \bigoplus_{i=1..n} y'[i] \end{aligned}$$

4.4 Security Analysis

In this section, we show that Bastion is mathrmind secure and (n – 2)CAKE secure.

LEMMA 1. Bastion is ind secure.

PROOF 1. Bastion uses an ind secure encryption mode to encrypt a message, and then applies a linear transform on the ciphertext blocks. It is straightforward to conclude that Bastion is ind secure. In other words, a polynomial-time algorithm A that has non-negligible advantage in breaking the ind security of Bastion can be used as a black-box by another polynomial-time algorithm B to break the ind security of the underlying encryption mode. In particular, B forwards A's queries to its oracle and applies the linear transformation of Algorithm 1 lines 7-14 to the received ciphertext before forward-ing it to A. The same strategy is used when A outputs two messages at the end of the find stage: the two messages are forwarded to B's oracle; upon receiving the challenge ciphertext, B applies the linear transformation and forwards it to A. When A replies with its guess b', B outputs the same guess. It is easy to see that if A has non-negligible advantage in guessing correctly which message was encrypted, so does B. Furthermore, the running time of B is the one of A plus the time to apply the linear transformation to A's queries.

LEMMA 2. Given any n – 2 blocks of y[1] . . . y[n] as output by Bastion, it is infeasible to compute any y'[i], for 1 ≤ i ≤ n.

PROOF 2. Let y = y[1], . . . , y[n] ← E(K, x = x[1] . . . x[m]). Note that given any (n – 1) blocks of y, the adversary can compute one block of y'. In particular, y'[i] = j=n y[j], for any 1 ≤ i ≤ n.
j=1,j6=i

As it will become clear later, with one block y'[i] and
L
the encryption key, the adversary has non-negligible probability of winning the game of Definition 3.

However, if only (n – 2) blocks of y are given, then each of the n blocks of y' can take on any possible values in {0, 1}l, depending on the two unknown blocks of y. Recall that each block y'[i] is dependent on (n – 1) blocks of y and it is pseudo-random as output by the CTR encryption mode. Therefore, given any (n – 2) blocks of y, then y'[i] could take any of the 2l possibilities, for 1 ≤ i ≤ n.

LEMMA 3. Bastion is (n – 2)CAKE secure.

PROOF 3. The security proof of Bastion resembles the standard security proof of the CTR encryption mode and relies on the existence of pseudo-random permutations. In particular, given a polynomial-type algorithm A which has non-negligible advantage in the (n – λ)CAKE experiment with λ = 2, we can construct a polynomial-time algorithm B which has non-negligible advantage in distinguishing between a true random permutation and a pseudo-random permutation.

B has access to oracle O and uses it to answer the encryption and decryption queries issued by A. In particular, A's queries are answered as follows:

2) Compute y'[i] = y[i] ⊕ t, for 1 ≤ i ≤ n
3) Compute x[i] = y'[i] ⊕ O(y'[n] + i), for 1 ≤ i ≤ n – 1

4) Return x[1] . . . x[n – 1]

- Encryption query for x[1] . . . x[n – 1]

1) Pick random y'[n] ∈ {0, 1}l

2) Compute y'[i] = x[i] ⊕ O(y'[n] + i), for 1 ≤i ≤ n – 1
3) Compute t = y'[1] ⊕ . . . ⊕ y'[n]
4) Compute y[i] = y'[i] ⊕ t, for 1 ≤ i ≤ n
5) Compute t = yb'[1] ⊕ . . . ⊕ yb'[n]
6) Compute yb[i] = yb'[i] ⊕ t, for 1 ≤ i ≤ n

At this point, A selects (n – 2) indexes i1, . . . in–2 and B returns the corresponding yb[i1], . . . , yb[in–2]. Encryption and decryption queries are answered as above. When A outputs its answer b', B outputs 1 if b = b', and 0 otherwise. It is straightforward to see that if A has advantage larger than negligible to guess b, then B has advantage larger than negligible to distinguish a true random permutation from a pseudorandom one. Furthermore, the number of queries issued by B to its oracle amounts to the number of encryption and decryption queries issued by A. Note that by Lemma 2, during the guess stage, A cannot issue a decryption query on the challenge ciphertext since with only (n – 2) blocks, finding the remaining blocks is infeasible.

7) Return y[1] . . . y[n]

When A outputs two messages x1[1] . . . x1[n–1] and x2[1] . . . x2[n – 1], B picks b ∈ {0, 1} at random and does the following:

1) Pick random yb'[n] ∈ {0, 1}l
2) Compute yb'[i] = xb[i]⊕O(yb'[n], i), for 1 ≤ i ≤ n–1

REMARK 6. Bastion is not (n – 1)CAKE secure. As shown in the proof of Lemma 2, the adversary can recover one block of y' given any (n – 1) blocks of y. If the adversary recovers y'[n] that is used as an IV in the CTR encryption mode, the adversary can easily win the (n – 1)CAKE game. Recall that our security definition allows the adversary to learn the encryption key.

REMARK 7. Bastion is (n – 2)CAKE secure according to Definition 3. However, in a practical deployment, we expect that each file spans several thousand blocks 5. When those blocks are evenly spread across servers, each server will store a larger number of blocks. Therefore, an (n – 2) CAKE secure scheme such as Bastion clearly preserves data confidentiality unless all servers are compromised.

V. CONCLUSIONS

In this paper, we tended to the issue of securing information outsourced to the cloud against an enemy, which approaches the encryption key. For that reason, we presented a novel security definition that catches information privacy against the new adversary. We at that point proposed Bastion, a plan that guarantees the classification of encoded information notwithstanding when the enemy has the encryption key, and everything except two figure content blocks. Bastion is most appropriate for settings where the cipher text blocks are put away in multi-distributed storage frameworks. In these settings, the foe would need to get the encryption key and to bargain all servers, keeping in mind the end goal to recoup any single piece of plaintext. We broke down the security of Bastion and assessed its execution in sensible settings. Bastion consider-capably enhances (by over half) the execution of existing natives which offer practically identical security under key presentation, and just acquires an irrelevant overhead (under 5%) when contrasted with existing semantically secure encryption modes (e.g., the CTR encryption mode). Finally, we demonstrated how Bastion could be essentially coordinated inside existing scattered stockpiling frameworks.

VI. REFERENCES

[1]. M. Abd-El-Malek, G. R. Ganger, G. R. Goodson, M. K. Re-iter, 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 doubly-iterated, ideal ciphers,"in Advances in Cryptology (CRYPTO), 1998, pp. 390–407.

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

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

[6]. A. Bessani, M. Correia, B. Quaresma, F. Andre, and P. Sousa, "DepSky: Dependable and Secure Storage in a Cloud-of-clouds,"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 All-or-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: Protect-ing against exhaustive key search,"in Advances in Cryptology (CRYPTO), 2000, pp. 359–375.

[13]. C. Dubnicki, L. Gryz, L. Heldt, M. Kaczmarczyk, W. Kil-ian, P. Strzelczak, J. Szczepkowski, C. Ungureanu, and

[14]. M.Welnicki, "HYDRAstor: a Scalable Secondary Storage,"in USENIX Conference on File and Storage Technologies (FAST), 2009, pp. 197–210.

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

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

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

[18]. J. Kilian and P. Rogaway, "How to protect DES against exhaustive key search,"in Advances in Cryptology (CRYPTO), 1996, pp. 252–267.

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

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

[21]. 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 Archi-tecture for Global-Scale Persistent Storage,"in International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2000, pp. 190–201.

[22]. L. Lamport, "On interprocess communication,"1985.

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

[24]. NEC Corp., "HYDRAstor Grid Storage,"http://www. hydrastor.com.

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

[26]. 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.

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

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

[29]. D. R. Stinson, "Something About All or Nothing (Trans-forms),"in Designs, Codes and Cryptography, 2001, pp. 133– 138.

[30]. StorSimple, "Cloud Storage,"http://www.storsimple.com/.

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

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

[33]. Z. Wu, M. Butkiewicz, D. Perkins, E. Katz-Bassett, and H. V. Madhyastha, "SPANStore: Cost-effective Geo-replicated Stor-age Spanning Multiple Cloud Services,"in ACM Symposium on Operating Systems Principles (SOSP), 2013, pp. 292–308.

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

AUTHOR DEATILAS

K. SANDHYA RANI is working an assistant professor in VIGNAN'S LARA INSTITUTE OF TECHNOLOGY & SCIENCE... Vadlamudi-522213 Guntur Dist. She has Experience in the teaching field For 2 years and her interested in research area data mining.

KOMMALAPATI RAJESH he is Currently pursuing MCA in MCA Department,Vignan's Lara Institute Of Technology&Science,Vadlamudi, Guntur, Andhra Pradesh, India.

He received his Bachelor of science from ANU