International Journal of Trend in Scientific Research and Development (IJTSRD)
International Open Access Journal
ISSN No: 2456 - 6470 | www.ijtsrd.com | Volume - 2 | Issue – 2

An Efficient Securing Code Based Cloud Storage using RC5 Encryption Algorithm against Pollution Attacks

Vani. S
Department of CSE, Mailam Engineering College, Mailam, Tamil Nadu, India

Priya Radhika Devi. T
Professor & Head, Department of CSE, Mailam Engineering College, Mailam, Tamil Nadu, India

ABSTRACT

The widespread diffusion of distributed and cloud storage solutions has changed dramatically the way users, system designers, and service providers manage their data. Outsourcing data on remote storage provides indeed many advantages in terms of both capital and operational costs. To provide solutions for the above pollution attack and user data breach, we propose an early pollution detection algorithm able to spot the presence of an attack while fetching the data from cloud storage during the normal disk reading operations. Thus each fragment is check for malicious content and passed over to the server for processing. The alarm triggers a procedure that locates the polluting nodes in early stage of processing to avoid the intrusion. Also to detect the polluted content hashing methods are used. In this way, after network peers receive the transmitted data, hash value is calculated by hash functions and then the result is compared with the received hash value from the content distribution network (CDN) servers to determine whether the transmitted data is polluted or not. I proposed system to addresses in preserving user data confidentiality and integrity after outsourcing. For this concern, we present our design for a new cloud storage encryption scheme that enables cloud storage providers to create **convincing fake** user secrets to protect user privacy.

I. INTRODUCTION

Cloud computing is a type of Internet-based computing that provides shared computer processing resources and data to computers and other devices on demand. It is a model for enabling ubiquitous, on-demand access to a shared pool of configurable computing resources (e.g., computer networks, servers, storage, applications and services), which can be rapidly provisioned and released with minimal management effort. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers that may be located far from the user–ranging in distance from across a city to across the world. Cloud computing relies on sharing of resources to achieve coherence and economy of scale, similar to a utility (like the electricity grid) over an electricity network.

Advocates claim that cloud computing allows companies to avoid upfront infrastructure costs (e.g., purchasing servers). As well, it enables organizations to focus on their core businesses instead of spending time and money on computer infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables Information Technology (IT) teams to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud providers typically use a "pay as you go" model. This will lead to unexpectedly high charges if administrators do not adapt to the cloud pricing model. Managing a private cloud requires software tools to help create a virtualized pool of compute resources, provide a self-service portal for end users and handle security, resource allocation, tracking and

billing. Management tools for private clouds tend to be service driven, as opposed to resource driven, because cloud environments are typically highly virtualized and organized in terms of portable workloads.

In hybrid cloud environments, compute, network and storage resources must be managed across multiple domains, so a good management strategy should start by defining what needs to be managed, and where and how to do it. Policies to help govern these domains should include configuration and installation of images, access control, and budgeting and reporting. Access control often includes the use of Single sign-on (SSO), in which a user logs in once and gains access to all systems without being prompted to log in again at each of them. A cloud management system combines software and technologies in a design for managing cloud environments. Software developers have responded to the management challenges of cloud computing with cloud management systems. HP, VMware, Red Hat, Novell, Eucalyptus, OpenNebula and Citrix, for example, sell management systems specifically for managing cloud environments. Cloud computing is an information technology (IT) paradigm, a model for enabling ubiquitous access to shared pools of configurable resources (such as computer networks, servers, storage, applications and services), which can be rapidly provisioned with minimal management effort, often over the Internet. Cloud computing allows users and enterprises with various computing capabilities to store and process data either in a privately-owned cloud, or on a third-party server located in a data center - thus making data-accessing mechanisms more efficient and reliable. Cloud computing relies on sharing of resources to achieve coherence and economy of scale, similar to a utility.

Advocates note that cloud computing allows companies to avoid or minimize up-front IT infrastructure costs. As well, third-party clouds enable organizations to focus on their core businesses instead of expending resources on computer infrastructure and maintenance. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and that it enables IT teams to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud providers typically use a "pay-as-you-go" model. This could lead to unexpectedly high charges if administrators are not familiarized with cloud-pricing models.

II. LITERATURE SURVEY

L. Buttyan, L. Czap, and I. Vajda, in this paper We address the problem of pollution attacks in coding-based distributed storage systems. In a pollution attack, the adversary maliciously alters some of the stored encoded packets, which results in the We incorrect decoding of a large part of the original data upon retrieval. We propose algorithms to detect and recover from such attacks. In contrast to existing approaches to solve this problem, our approach is not based on adding cryptographic checksums or signatures to the encoded packets, and it does not introduce any additional redundancy to the system. The results of our analysis show that our proposed algorithms are suitable for practical systems, especially in wireless sensor networks.

Vander Alves, Nan Niu, Carina Alves, George Valença Network coding is promising to maximize throughput in various networking systems. Compared to normal network coding operated over large finite fields, XOR network coding has gained an increasing number of applications for its simplicity, especially in wireless networks. However, both types of network coding systems are vulnerable to pollution attacks in which the compromised forwarders inject polluted messages into the systems. Existing solutions to pollution attacks can protect only the normal network coding, but none of them is able to secure XOR network coding. In this paper, we propose an efficient scheme for securing XOR network coding against pollution attacks. Our scheme exploits probabilistic key pre-distribution and message authentication codes (MACs). In our scheme, the source appends multiple MACs to each message, where each MAC can authenticate only a part of the message and the parts authenticated by different MACs are overlapped. Thus, multiple forwarders can collaboratively verify different parts of messages using the MACs with their own shared keys. By carefully controlling the overlapping between the parts authenticated by different MACs, our scheme can filter polluted messages in a few hops with a high probability. To the best of our knowledge, this is the first solution to pollution attacks for XOR network coding. Experimental results show that it is 200 to 1000 times faster than existing ones, hence, it is particularly suitable for resource-constrained wireless networks.

L. Buttyn, L. Czap, and I. Vajda We present a novel information theoretic approach to make network coding based storage secure against pollution attacks in sensor networks. The approach is based on a new decoding algorithm which makes it possible to find adversarial blocks using one more encoded block than strictly necessary for decoding. Our scheme fits well to the requirements of sensor networks, because it operates with adding very low computational and communication overhead to source and storage nodes, only the collector node needs to perform some additional computation. Our approach does not apply cryptography, hence it works in environments where no pre-shared keys, secure channels or PKI are available, which is often the case in sensor networks.

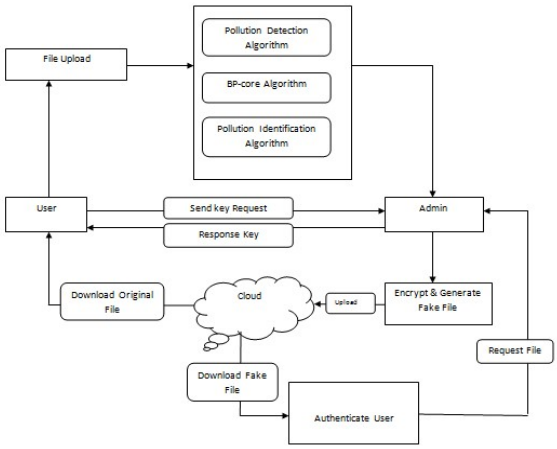

Fig: Overall Architecture

III. PROPOSED SYSTEM

To provide solutions for the above pollution attack and user data breach, we propose an early pollution detection algorithm able to spot the presence of an attack while fetching the data from cloud storage during the normal disk reading operations. Thus each fragment is check for malicious content and passed over to the server for processing. The alarm triggers a procedure that locates the polluting nodes in early stage of processing to avoid the intrusion. Also to detect the polluted content hashing methods are used. In this way, after network peers receive the transmitted data, hash value is calculated by hash functions and then the result is compared with the received hash value from the content distribution network (CDN) servers to determine whether the transmitted data is polluted or not.

Our proposed system also addresses in preserving user data confidentiality and integrity after outsourcing. For this concern, we present our design for a new cloud storage encryption scheme that enables cloud storage providers to create convincing fake user secrets to protect user privacy. Since coercers cannot tell if obtained secrets are true or not, the cloud storage providers ensure that user privacy is still securely protected. For encryption we have proposed Audit free cloud algorithm.

USER REGISTRATION AND VALIDATION

Data validation is the process of ensuring that a program operates on clean, correct and useful data. It uses routines, often called "validation rules" "validation constraints" or "check routines", that check for correctness, meaningfulness, and security of data that are input to the system. The rules may be implemented through the automated facilities

TRANSFORMING DATA INTO FRAGMENTS

A program transformation is any operation that takes a computer program and generates another program. In many cases the transformed program is required to be semantically equivalent to the original, relative to a particular formal semantics and in fewer cases the transformations result in programs that semantically differ from the original in predictable ways While the transformations can be performed manually, it is often more practical to use a program transformation system that applies specifications of the required transformations. Program transformations may be specified as automated procedures that modify compiler data structures representing the program text, or may be specified more conveniently using patterns representing parameterized source code text fragments.

POLLUTION ATTACK DETECTION

Pollution attack is defined as the injection of bogus coded fragments sent by malicious nodes in response to read requests, and thus in charge of reconstructing the original sectors starting from the set of corresponding coded fragments can be also used to carry out pollution detection. It is possible that a single polluted coded fragments propagates to many

original sector fragments. However, in this work we show that coding brings also significant benefits in terms of pollution detection, since it can be exploited to both detect pollution and identify the nodes responsible of the damage.

HASHING METHOD

In this hashing method we encrypt the given data and stored into the database, after if anyone ask the request to get the file it will check the hash value is same or not, then only it allow to view or download the data from database..

CONVINCING FILE CREATION

Most of the proposed schemes assume cloud storage service providers or trusted third parties handling key management are trusted and cannot be hacked; however, in practice, some entities may intercept communications between users and cloud storage providers and then compel storage providers to release user secrets by using government power or other means. In this case, encrypted data are assumed to be known and storage providers are requested to release user secrets. Since it is difficult to fight against outside coercion, we aimed to build an encryption scheme that could help cloud storage providers avoid this predicament. In our approach, we offer cloud storage providers means to create fake user secrets. Given such fake user secrets, outside coercers can only obtained forged data from a user's stored cipher text. Once coercers think the received secrets are real, they will be satisfied and more importantly cloud storage providers will not have revealed any real secrets. Therefore, user privacy is still protected.

RIVEST CIPHER ENCRYPTION

The RC5 encryption algorithm is a fast, symmetric block cipher suitable for hardware or software implementations. A novel feature of RC5 is the heavy use of data-dependent rotations. RC5 has a variable-length secret key, providing flexibility in its security level. The algorithm can be broken into two stages: initialization, and operation.

In the initialization stage the 256-bit state table, S is populated, using the key, K as a seed. Once the state table is setup, it continues to be modified in a regular pattern as data is encrypted. The initialization process can be summarized by the pseudo-code;

```
j = 0;

for i = 0 to 255:

S[i] = i;

for i = 0 to 255:

j = (j + S[i] + K[i]) mod 256;

swap S[i] and S[j];
```

It is important to notice here the swapping of the locations of the numbers 0 to 255 (each of which occurs only once) in the state table. The values of the state table are provided. Once the initialization process is completed, the operation process may be summarized as shown by the pseudo code below;

```
i = j = 0;

for (k = 0 to N-1) {

i = (i + 1) mod 256;

j = (j + S[i]) mod 256;

swap S[i] and S[j];

pr = S[ (S[i] + S[j]) mod 256]

output M[k] XOR pr }
```

Where M[0..N-1] is the input message consisting of N bits.

RC Encryption Steps:

The steps for RC4 encryption algorithm is as follows:

1- Get the data to be encrypted and the selected key.

2- Create two string arrays.

3- Initiate one array with numbers from 0 to 255.

4- Fill the other array with the selected key.

5- Randomize the first array depending on the array of the key.

6- Randomize the first array within itself to generate the final key stream.

7- XOR the final key stream with the data to be encrypted to give cipher text.

AUDIT FREE CLOUD STORGE

Public cloud storage is a cloud storage model that enables individuals and organizations alike to store, edit and manage data. This type of storage exists on a remote cloud server and is accessible over the Internet. Public cloud storage is provided by a storage service provider that hosts, manages and sources the storage infrastructure publicly to many different users. Public cloud storage service is also known as storage as a service, utility storage and online storage. For this project we are trying to use Sync public cloud.

IV. SYSTEM IMPLEMENTATION

DETECTION ALGORITHM

Rateless coding can be used to increase sector availability since the same original information can be retrieved from any random set of more than k coded fragments. At a first glance, it could seem that the use of coding can make the system very vulnerable to pollution. Indeed, as shown in Fig. 1, it is possible that a single polluted coded fragments propagates to many original sector fragments si. However, in this work we show that coding brings also significant benefits in terms of pollution detection, since it can be exploited to both detect pollution and identify the SNs responsible of the damage. To this end, we need to look in more detail at the LT decoding process. LT decoding can be cast as the solution of a linear system of equations GS = F where G is a k×k decoding matrix carrying on the rows k linearly independent coding vectors, F is the column vector of the k coded fragments corresponding to such coding vectors, and S is the vector of k unknown original sector fragments. In the following, we denote as Gl the l-th row of G and Fl the l-th element of vector F.

The OFG decoder sequentially processes the input, i.e., the pairs (fi, gi), that are being provided by SNs, and executes Gaussian Elimination on the fly to progressively fill up G starting from an empty matrix, until G turns full rank, that corresponds to the recovery of the original sector. A modified version of the OFG decoder, that includes the pollution detection mechanism we propose in this paper is shown in Alg. 1 below by using pseudo-code. The algorithm takes as input a set of coded fragments Q, processes the coded fragments (f, g) 2 Q, and returns row insertion process will not be affected by pollution.

When processing the pair (fp, g) two outcomes are possible:

a) g is linearly independent on current G. In this case any polluted fragment is stored in a given row of G and F (lines 6-8 in Alg. 1); b) g is linearly dependent on current G. In this case the decoder knows a linear combination Pk j=1 _jFj that should coincide with fp. If all previous rows are clean, it turns out that Pk j=1 _jFj = f 6= fp and therefore pollution is detected (lines 12-14 in Alg. 1). If polluted equations were inserted before (see previous item a) one gets Pk j=1 _jFj = f +Pk

j=1 _jrj , that is a combination of random pollution patterns (assuming rj = 0 for clean equations).

Algorithm 1 Decode(Q)

1: decoded = **false**, polluted = **false**

2: **for all** (f, g) 2 Q **do**

3: **while true do**

4: {Gaussian Elimination}

5: l position of leading one of g.

6: **if** Gl = ; **then**

7: Gl g; Fl f

8: **break**

9: **else**

10: **if** g = Gl **then**

11: {g is linearly dependent on current G}

12: **if** f 6= Fl **then**

13: polluted = **true**

14: **end if**

15: **break**;

16: **else**

17: {g is linearly independent on current G}

18: g g + Gl; f f + Fl

19: **end if**

20: **end if**

21: **end while**

22: **end for**

23: **if** G is full rank **then**

24: decoded = **true**

25: **end if**

REFERENCES

1. N. Cao, S. Yu, Z. Yang, W. Lou, and Y. Hou, "LT codes-based secure and reliable cloud storage service," in *IEEE INFOCOM*, 2012, pp. 693–701.
2. L. Buttyn, L. Czap, and I. Vajda, "Pollution attack defense for coding based sensor storage," in *Proceedings of the IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC)*, 2010.
3. M. N. Krohn, M. J. Freedman, and D. Mazieres, "On-the-fly verification of rateless erasure codes for efficient content distribution," *Security and Privacy, IEEE Symposium on*, 2004.
4. C. Gkantsidis and P. Rodriguez, "Cooperative security for network coding file distribution," in *IEEE INFOCOM*, 2006.
5. Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature based scheme for securing network coding against pollution attacks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008.
6. E. Kehdi and B. Li, "Null keys: Limiting malicious attacks via null space properties of network coding," in *INFOCOM 2009, IEEE*.
7. Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient scheme for securing xor network coding against pollution attacks," in *INFOCOM 2009, IEEE*.
8. X. Wu, Y. Xu, C. Yuen, and L. Xiang, "A tag encoding scheme against pollution attack to linear network coding," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 33–42, 2014.
9. F. Chen, T. Xiang, Y. Yang, and S. Chow, "Secure cloud storage meets with secure network coding," in *INFOCOM, 2014 Proceedings IEEE*, pp. 673–681.
10. A. Le and A. Markopoulou, "Nc-audit: Auditing for network coding storage," in *Network Coding (NetCod), 2012 International Symposium on*, pp. 155–160.