iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** III **Month of publication:** March 2018

DOI: http://doi.org/10.22214/ijraset.2018.3477

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

Data Blurt in Unsecured Environment

P. Haritha[1], MD. Matheen[2], J. Chaitanya Pavan[3], S. Teja Sai[4], K. Rajasekhar[5]
[1, 2, 3, 4, 5,] *Department of IT, LBRCE*

Abstract: ***The word "Data Transfer" has become a twilight now a days. In this particular process of transferring data we face some issues of which leakage of confidential data is the most severe threat faced by almost all organisations in modern age. There is a huge amount of personal information available on social networks and Smartphone providers, which is urged to transfer from one source to another. Which helps users to enjoy latest when this type of confidential data is being transferred, knowingly or unknowingly the data may be transferred to entrusted third party or fourth party applications .In our work, we present a generic data lineage in malicious framework named "LIME" which demonstrates the flow of data across multiple entities that take two featured principal roles. Identification of a culpable entity and honest assumptions is done by defining the exact security guarantees. An accountable data transfer protocol between two entities within unsecured environment is developed and analysed. Finally an experimental evaluation is performed to demonstrate the practicality of our protocol.***
Keywords: Data transfer, leakage, malicious, LIME, culpable entity

I. INTRODUCTION

In the digital era, information leakage through accidental acquaintances, or intentional deliberately destroyed by dissatisfied employees and mischievous external entities, present one of the most serious threats to organizations. according to an interesting chronology of data breaches maintained by the privacy rights clearinghouse .it is not hard to believe that this is just the tip of the iceberg, as most cases of information leakage is unrecorded due to fear of loss of customer confidence or regulatory penalties: it costs companies on average $214 per compromised record [1] .hefty amounts of digital data can be copied at almost no cost and can be spread through the internet in very short time. Additionally, the risk of getting caught for data blurt is very low, as there are currently almost no accountability mechanisms. for these reasons, the problem of data leakage has reached new proportions nowadays. Not only companies are affected by data leakage, it is also a concern to individuals. the rise of social networks and smart phones has made the situation more worse. In these environments, individuals make known their personal information to various service providers, commonly known as third party applications, in return for some possibly free services. in the absence of proper regulations and accountability mechanisms, many of these applications share individuals' identifying information with dozens of advertising and internet tracking companies. Even with access control mechanisms, where access to sensitive data is limited, a malicious authorized user can publish sensitive data as soon as he receives it. primitives like encryption offer protection only as long as the information of interest is encrypted, but once the recipient decrypts a message, nothing can prevent him from publishing the decrypted content. Thus it seems impossible to prevent data leakage aggressively.

Privacy, consumer rights, and advocacy organizations such as prc [2] and epic [3] try to address the problem of information leakages through policies and awareness. However, as seen in the following scenarios the effectiveness of policies is questionable as long as it is not possible to provably associate the guilty parties to the leakages. In this data blurt we are having two scenarios. Scenario 1: social networking. it was reported that third party applications of the widely used online social network face book leak sensitive private information about the users or even their friends to advertising companies [4] in this case, it was possible to determine that several applications were leaking data by analysing their behaviour and so these applications could be disabled by face book. However, it is not possible to make a particular application responsible for leakages that already happened, as many different applications had access to the private data. Scenario 2: outsourcing. Florida state employees were informed that their personal information has been compromised due to improper outsourcing [5]. the outsourcing company that was handed sensitive data hired a further subcontractor. Although the offshore subcontractor is suspected, it is not possible to provably associate one of the three companies to the leakage, as each of them had access to the data and could have possibly leaked it.

We find that the above and other data leakage scenarios can be associated to an absence of accountability mechanisms during data transfers: leakers either do not focus on protection, or they intentionally expose confidential data without any concern, as they are convinced that the leaked data cannot be linked to them. in other words, when entities know that they can be held accountable for leakage of some information, they will demonstrate a better commitment towards its required protection. in some cases, identification of the leaker is made possible by forensic techniques, but these are usually expensive and do not always generate the desired results. therefore, we point out the need for a general accountability mechanism in data transfers. this accountability can be

directly associated with *probably*detecting a transmission history of data across multiple entities starting from its origin. this is known as data origin, data blurt or data source tracing. the data provenance methodology, in the form of robust watermarking techniques or adding fake data has already been suggested in the literature and employed by some industries. however, most efforts have been ad-hoc in nature and there is no formal model available. Additionally, most of these approaches only allow identification of the leaker in a non-provable manner, which is not sufficient in many cases.

II. RELATED WORK

[6]A data trader has given sensitive data to a set of apparently trusted agents (third parties). Some of the data is leaked and found in an unauthorized place (e.g., on the web or somebody's laptop). The trader must assess the likelihood that the blurted data came from one or more agents. We propose data allocation strategies (across the agents) that improve the probability of identifying leakages. Its goal is to detect when the trader own sensitive data that has been blurted by agents, and if possible we can also identify the agent that blurted the data. Perturbation is a very useful technique where the data is modified and made the quote less sensitive quote before being handed to agents. We develop attractive techniques for detecting blurt of a set of objects or records or data. Here a model for assessing the less quote data for agents is developed. Here algorithms for trading objects to agents are presented which helps in improving the chance of identifying a leaker. In perturbation method the exact value of the sensitive data can be replaced by ranges, rounded off to the nearest integer to find the guilt agents, this may be a big problem for confidential data. [7]Multimedia security schemes often combine cryptographic schemes with information hiding techniques such as steganography or watermarking. Example applications are dispute resolving, proof of ownership, (asymmetric/anonymous) fingerprinting and zero-knowledge watermark detection. The need for formal security definitions of watermarking schemes is manifold, whereby the core need is to provide suitable abstractions to construct, analyse and prove the security of applications on top of watermarking schemes. Although there exist formal models and definitions for information-theoretic and computational security of cryptographic and stenographic schemes, they cannot simply be adapted to watermarking schemes due to the fundamental differences among these approaches. Moreover, the existing formal definitions for watermark security still suffer from conceptual deficiencies.

III. EXISTING SYSTEM

In previous system we are confined only to a single provider. The disadvantage in that system is the provider always cannot provide better resources and services for end user. Cloud Broker is not acting as middleware between cloud provider and user so that security and reliability of service is not guaranteed.

IV. PROPOSED WORK

In the present system we are using AES algorithm to encrypt and decrypt the data and RSA algorithm is used to generate the secret key and digital signature. In this system we can also detect the data leakages; we can find the attacker who attacked the data within the organisation intentionally or unintentionally and In general case of data leakage in data transfer settings, we propose more comprehensible model LIME (Lineage in the malicious environment). With LIME we assign a clearly defined role to each involved party and define the inter-relationships between these roles. In this work four modules are used to develop this system a) Data Owner Module b) Data Consumer Module c) Web Server Module d) Auditor Module.

V. IMPLEMENTATION

In this system, in order to get the fruitful results we use two algorithms and a framework.

A. Data Owner Module

In this module, the data owner uploads their Images with their contents data to the Web server. For the security purpose the data owner assigns the digital sign and then store in the Web and also performs the following operations such as Upload image with its digital sign based on title, description, List all uploaded images, Verify image details, and Delete image details.

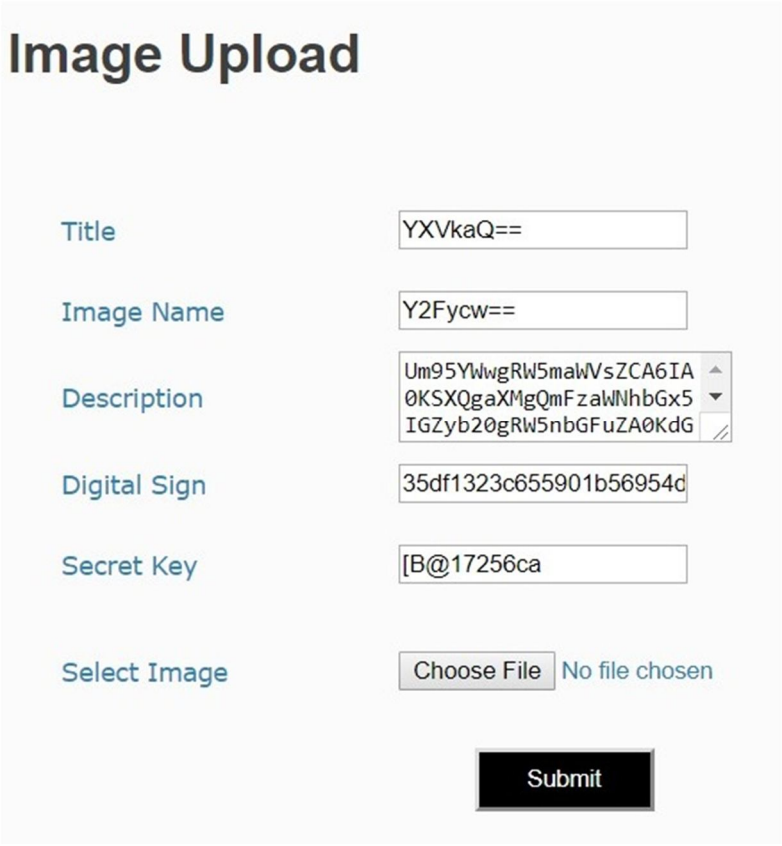


Fig 5.1.1 represents generation of digital sign and secret key when image or file is uploaded

B. Data Consumer Module

The Web User who has a large amount of data to be stored in Web Servers and have the permissions to access and manipulate stored image and its data. The consumer will search the data and accessing the image data if he is authorized and performs the following operations such as Search image, Access image and its details, Download image & make common.

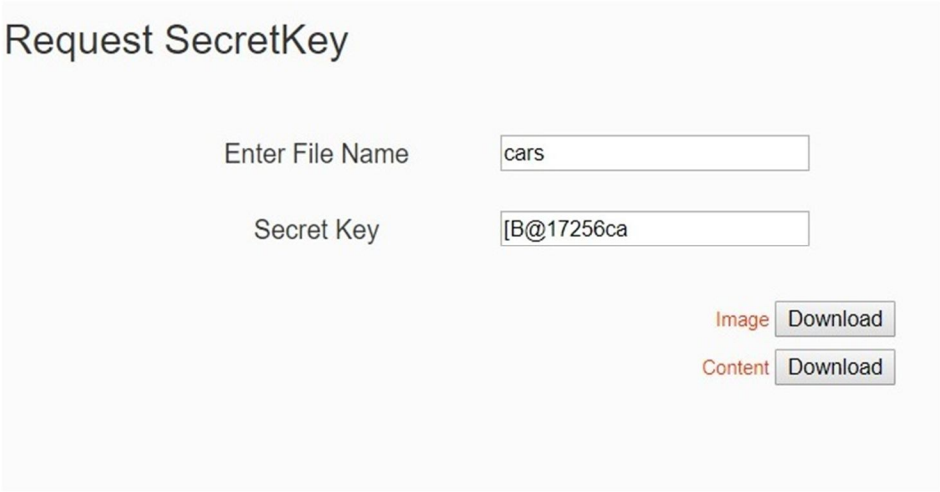


Fig 5.2.1 represents generation of secret key for consumer to download the file or image

C. Web Server Module

The Web service provider manages a Web to provide data storage service. And performs the following operations such as Store all image files with their signature, View all image Files with its details, View all image comments, View all Data owners and Users, View all attackers.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

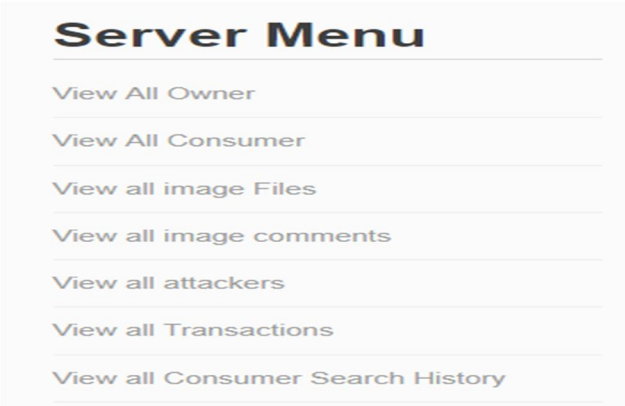


Fig 5.3.1 represents sever actions

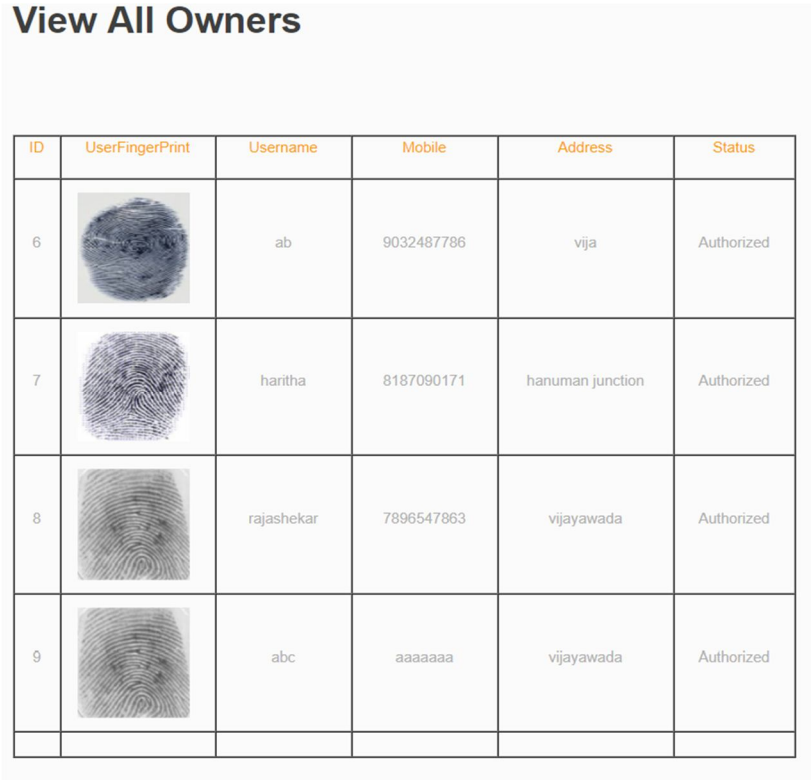


Fig 5.3.2 represents one of the actions performed by the server

D. Auditor Module

Auditor who has capabilities to manage or monitor the outsourced data under the delegation of data owner, for periodically auditing the outsourced data. This audit service is significantly important for digital forensics and credibility in Webs and performs the following operations such as Store all Meta data of the images, View all Image metadata such as image id, image title, and Digital sign, Viewall

VI. LIME FRAME WORK

In general case of data leakage in data transfer settings, we propose more comprehensible model LIME (Lineage in the malicious environment). With LIME we assign a clearly defined role to each involved party and define the inter-relationships between these roles. This allows us to define the exact properties that our transfer protocol has to fulfil in order to allow a provable identification of the guilty party in case of data leakage.When documents are transferred from one owner to another one, we can assume that the transfer is governed by a non-repudiation assumption. This means that the sending owner trusts the receiving owner to take responsibility if he should leak the document. As we consider consumers as entrusted participants in our model, a transfer involving

a consumer cannot be based on a non-repudiation assumption. Therefore, whenever a document is transferred to a consumer, the sender embeds information that uniquely identifies the recipient. We call this fingerprinting. If the consumer leaks this document, it is possible to identify him with the help of the embedded information.

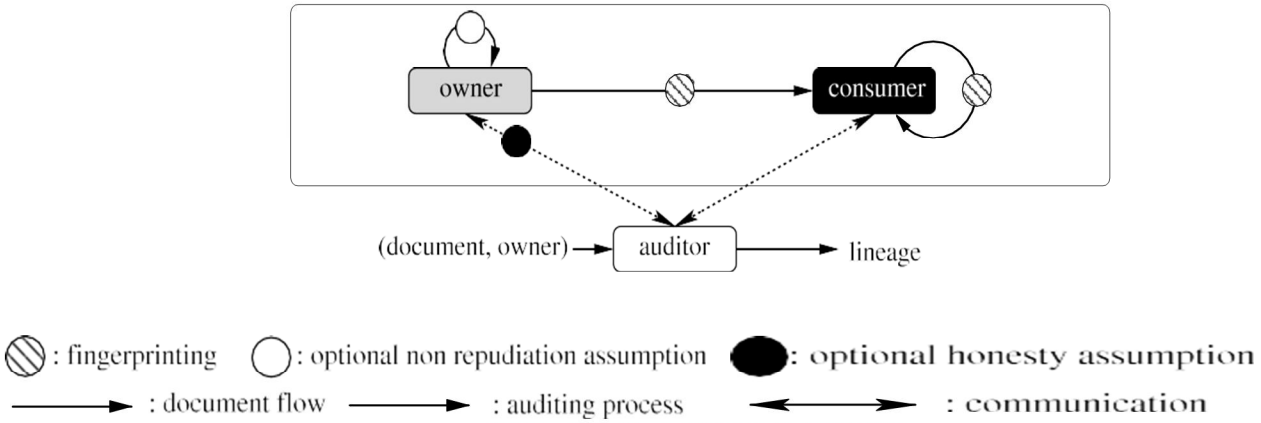


Figure 6.1: The LIME FRAMEWORK

The framed box shows document transfers between owners and consumers. The auditor is a special entity which is onlyrequired when a leakage occurs. The auditor then reconstructs the data lineage by communicating with the involved parties.

A. AES Algorithm

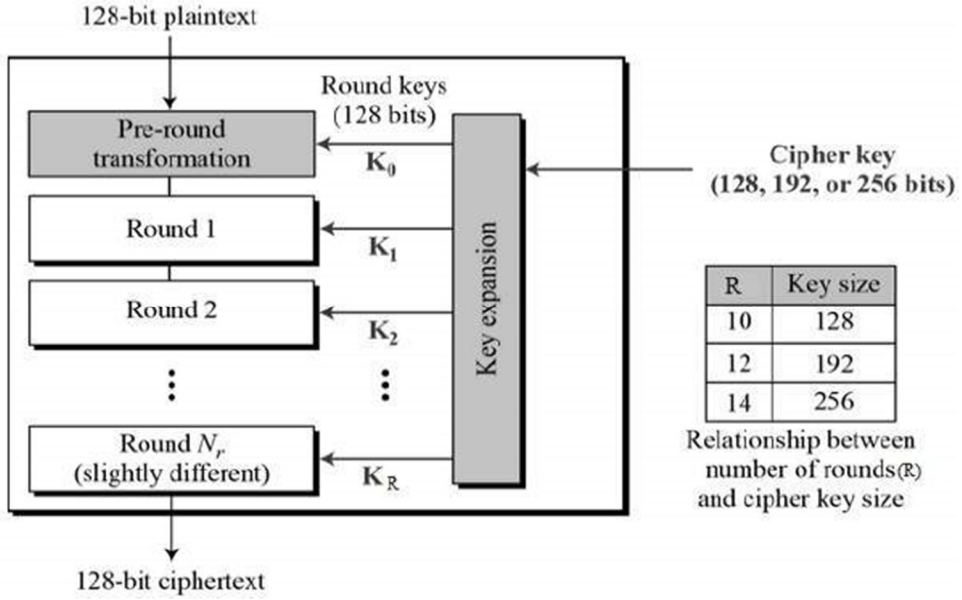
The main objective of this algorithm is to encrypt as well as decrypt the input given to it.

B. Operation of AES

AES is an iterative rather than Feistel cipher. It is based on 'substitution–permutation network'. It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).

Interestingly, AES performs all its computations on bytes rather than bits. Hence, AES treats the 128 bits of a plaintext block as 16 bytes. These 16 bytes are arranged in four columns and four rows for processing as a matrix –

Unlike DES, the number of rounds in AES is variable and depends on the length of the key. AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.The schematic of AES structure is –

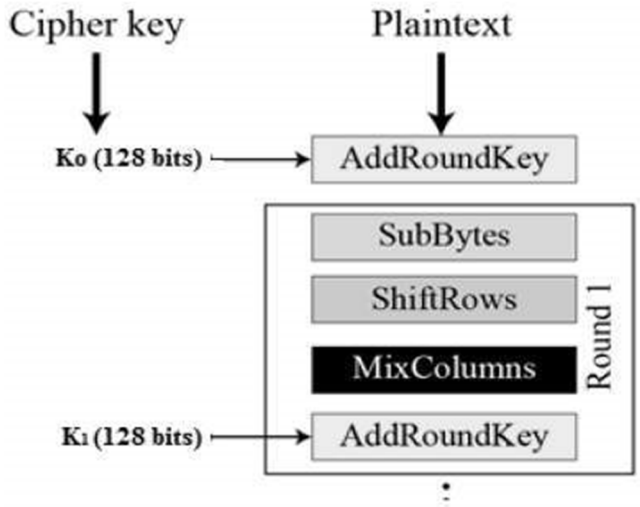


R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

C. Encryption Process

Here, we restrict to description of a typical round of AES encryption. Each round comprise of four sub-processes. The first round process is depicted below −



D. Byte Substitution (Sub Bytes)

The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.

E. Shift Rows

Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row. Shift is carried out as follows −

First row is not shifted.
Second row is shifted one (byte) position to the left.
Third row is shifted two positions to the left.
Fourth row is shifted three positions to the left.
The result is a new matrix consisting of the same 16 bytes but shifted with respect to each other.

F. Mix Columns

Each column of four bytes is now transformed using a special mathematical function. This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column. The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.

G. Add Round key

The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key. If this is the last round then the output is the ciphertext. Otherwise, the resulting 128 bits are interpreted as 16 bytes and we begin another similar round.

H. Decryption Process

The process of decryption of an AES ciphertext is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order −

Add round key
Mix columns
Shift rows
Byte substitution

Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms needs to be separately implemented, although they are very closely related.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue III, March 2018- Available at www.ijraset.com

I. RSA Algorithm

The main objective of this algorithm is to generate digital signature key and secret key which play a major role in uploading the data to server and downloading the data from the server i.e. when a file is uploaded to server it generates digital signature and secret key to that image or file.

J. Operation of RSA algorithm

Sequence of blocks M1, M2, . . . , Mt where each Mi satisfies $0 \leq Mi < n$. Then encrypt these blocks as

$$C = E(M) = Me \pmod{n}; \quad (1)$$

Decryption: Given the private key d and the cipher text C, the decryption function is:

$$D(C) = Cd \pmod{n}; \quad (2)$$

Note that encryption does not increase the size of a message. Both the message and the cipher text are integers in the range 0 to n - 1.

The encryption key is thus the pair of positive integers (e; n). Similarly, the decryption key is the pair of positive integers (d; n). Each user makes his encryption key public, and keeps the corresponding decryption key private.

K. Re-Attacking process

This the main process where the attacker is being identified. In a secure organisation there is a chance of leakage of secure data. The file or image which has been attacked will be notified that it is attacked but it doesn't have any proof that it is being done by a particular person. With the help of this re-attacking process the auditor or data owner may attack with the name of attacker or the data which he has near him with the help of notification given when it was attacked. If the data matches then we can easily know who the attacker is. In order to describe this re-attacking process we have used a framework to visualise it.

Fig 6.2 when a file is attacked by an attacker

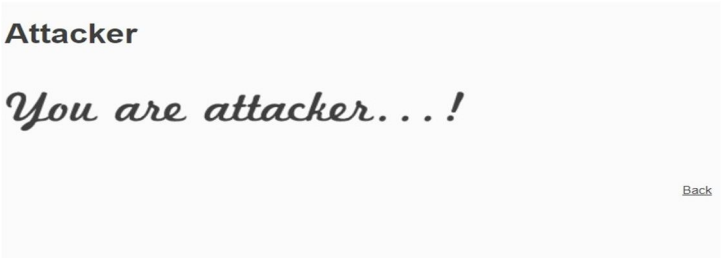


Fig 6.3 when the attacker is found with the help of re-attacking process

L. Architecture Diagram

This system we have four main pillarsi.e., data consumer, data owner, web server, and auditor they perform various operation to protect the data and also finds the attacker who attacked the file .This system is used within the organisation when any one of the employee thefts the data whichwe cannot blame any employee without proof. we can recognize the attacker by using re-attacking process. this system provides proof which is very useful to find the attacker. In this system first the data owner uploads the image or file into the system when that file or image is being uploaded into the system particular secret key and digital sign are given to generated to that file. Data owner also lists all the uploaded files, verifies the image details and also delete the image if it is necessary. Now the server stores the uploaded files from an authorised user along with the digital signature of that particular file or an image. The server also views all the image files with their details and also the server maintains the information of all data owners and attackers. The auditor maintains the meta data of a particular file or an image it also has all the relevant information of a

particular file such as image id, title etc. the consumer now searches a particular file access the image and downloads it. He also comment on that particular image when needed.

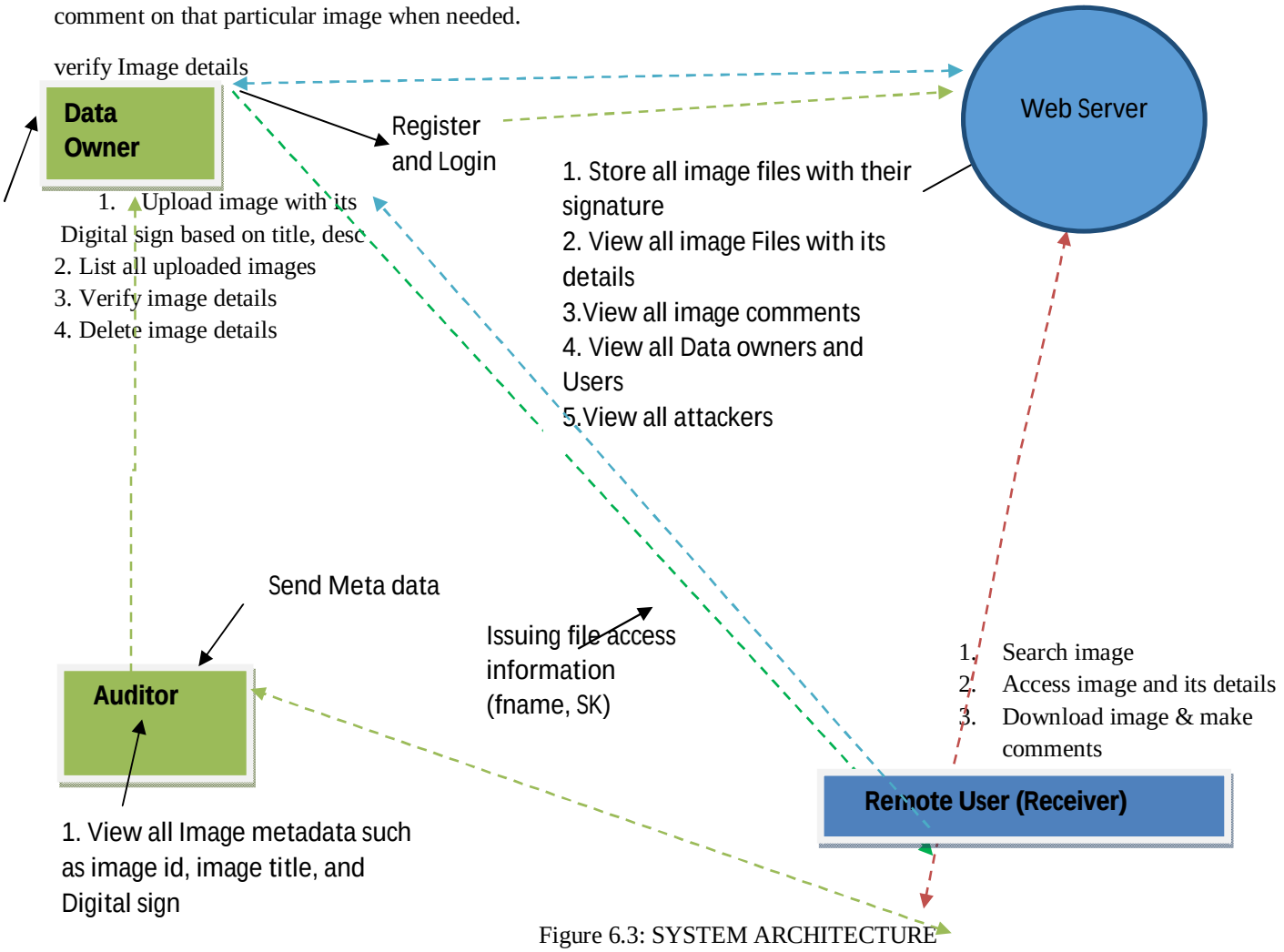


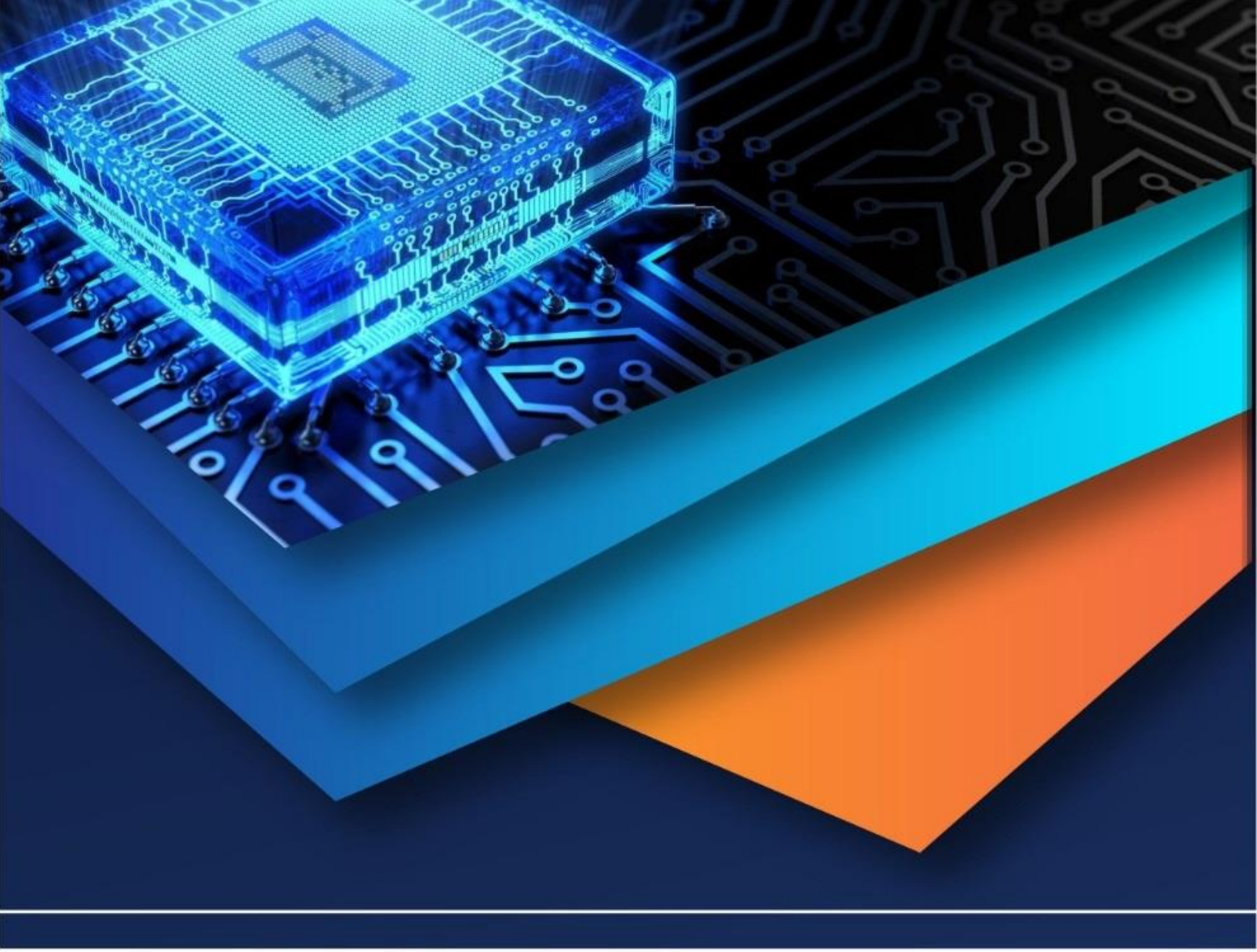
Figure 6.3: SYSTEM ARCHITECTURE

VII. CONCLUSION

We present a generic data lineage framework named "LIME" which demonstrates the flow of data across multiple entities that take two characteristics, principal roles. Identification of a guilty entity, and honest assumptions is done by defining the exact security guarantees. An accountable data transfer protocol between two entities within a malicious environment is developed and analysed. Finally, an experimental evaluation is performed to demonstrate the practicality of our protocol.

REFERENCES

[1] Data breach cost [Online]. Available: http://www.symantec.com/about/news/release/article.jsp?prid=20110308_01, 2011.
[2] Privacy rights clearinghouse [Online]. Available: http://www. privacyrights.org, 2014
[3] (1994). Electronic privacy information center (EPIC) [Online]. Available: http://epic.org, 199
[4] Facebook in privacy breach [Online]. Available: http://online.wsj.com/article/SB10001424052702304772804575558484075236968. html, 2010
[5] ://www.computerworld.com/s/article/109938/Offshore_outsourcing_cited_in_Florida_data_leak, 2006
[6] P. Papadimitriou and H. Garcia-Molina, "Data leakage detection," IEEE Trans. Knowl. Data Eng., vol. 23, no. 1, pp. 51–63, Jan. 2011
[7] A. Adelsbach, S. Katzenbeisser, and A.-R. Sadeghi, "A computational model for watermark robustness," in Proc. 8th Int. Conf. Inf. Hiding, 2007, pp. 145–160
[8] M. Backes, N. Grimm, and A. Kate, "Lime: Data lineage in the malicious environment," in Proc. 10th Int. Workshop Security Trust Manage., 2014, pp. 183–187
[9] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signaturescheme secure against adaptive chosen-message attacks," SIAMJ. Comput., vol. 17, no. 2, pp. 281–308, 1988.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY