IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 01-07

Provably Associating The Guilty Party To The Leakages

[1] Karthikeyan., [2]mrs.V.Vimala
V M.E Pg Student ,M.e., Assiciate Professor
Dept. Of Computer Science &Engg. [1,]2Skr Engineering College, Agarmel, Chennai-600 123.

Abstract: *Intentional or unintentional escape of confidential information is beyond questionone in every ofthe foremost severe security threats that organizations face within the digital era. The threat currently extends to our personal lives: a excessof non-publicinfois out there to social networks and smartphone suppliers and is indirectly transferred to teflon third party and fourth party applications. During this work, we have a tendency togift a generic information lineage framework LIME for information flow across multiple entities that take two characteristic, principal roles (i.e., owner and consumer). We have a tendency tooutlinethe precise security guarantees needed by such an information lineage mechanism toward identification of a guilty entity, and determine the simplifying non-repudiation and honesty assumptions. We have a tendency to then develop and analyze a completely uniqueresponsibleinformation transfer protocol between two entities at intervals a malicious setting by building upon oblivious transfer, strong watermarking, and signature primitives. Finally, we have a tendency to perform associate experimental analysis to demonstrate the usefulness of our protocol and apply our framework to the necessaryinformationescapesituationsof information outsourcing and social networks. In general, we have a tendency totake into account LIME, our lineage framework for information transfer, to be associate key step towards achieving responsiblenessdesignedly*.

I Introduction

Network security consists of the provisions and policies adopted by a network administrator to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator. Network security covers a variety of computer networks, both public and private, that are used in everyday jobs conducting transactions and communications among businesses, government agencies and individuals. Networks can be private, such as within a company, and others which might be open to public access. Network security is involved in organizations, enterprises, and other types of institutions. Web applications also present new security and privacy challenges, partly because the un-trusted Internet has essentially become an integral component of such applications for carrying the continuous interaction between users and servers.

Every industry has its own "language," the jargon that describes concepts and procedures peculiar to the field. Computer networking is infamous for the "techno talk" and the proliferation of acronyms that often mystify outsiders. Specialty areas within an industry often have their own brands of jargon, as well, and the computer security sub-field is no exception.

Threats can come from both internal and external sources. They may be human based, automated or even non intentional natural phenomenon. The latter might more appropriately be categorized under system health threats as opposed to security threats, but one issue can lead to the other.Trojan horse is a computer program that appears to perform a desirable function but contains hidden code that is intended to allow unauthorized collection, modification or destruction of data.Virusis a program that is introduced onto a system or network for the purpose of performing an unauthorized action (which can vary from popping up a harmless message to destroying all data on the hard disk).

Vulnerability is a weakness in the hardware or software or security plan that leaves a system or network open to threat of unauthorized access or damage or destruction of data.Worm is a program that replicates itself, spreading from one machine to another across a network. Once you are comfortable with the terminology, you can begin to address the individual objectives that will assist you in realizing your goal to create a secure network environment. The attack type refers to how an intruder gains entry to your computer or network and what he does once he has gained entry.

II Related Work

Data usage control enforcement in distributed systems proposed by F. Kelbert and A. Pretschner [1] is concerned with how data may or may not be used in distributed system environments after initial access has been granted. If data flows through a distributed system, there exist multiple copies of the data on different client machines. Usage constraints then have to be enforced for all these clients. We extend a generic model for

intra-system data flow tracking—that has been designed and used to track the existence of copies of data on single clients—to the cross-system case. When transferring, i.e., copying, data from one machine to another, our model makes it possible to transfer usage control policies along with the data to the end of local enforcement at the receiving end, and to be aware of the existence of copies of the data in the distributed system. As one example, we concretize "transfer of data" to the Transmission Control Protocol (TCP). Based on this concretized model, we develop a distributed usage control enforcement infrastructure that generically and application-independently extends the scope of usage control enforcement to any system receiving usagecontrolled data. The implemented work for OpenBSD and evaluated security and performance gives worst output.

Two main results in the area of information hiding in natural language text are presented by M. J. Atallah, V. Raskin, C. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg as natural .anguage watermarking and tamperproofing [2]. A semantically-based scheme dramatically improves the information-hiding capacity of any text through two techniques: (i) modifying the granularity of meaning of individual sentences, whereas our own previous scheme kept the granularity fixed, and (ii) halving the number of sentences affected by the watermark. No longer a "long text, short watermark" approach, it now makes it possible to watermark short texts like wire agency reports. Using both the above-mentioned semantic marking scheme and the previous syntactically-based method hides information in a way that reveals any non-trivial tampering with the text (while re-formatting is not considered to be tampering—the problem would be solved trivially otherwise by hiding a hash of the text) with a probability 1–2– (n+1), n being its number of sentences and a small positive integer based on the extent of co-referencing.

As network capacity has increased over the past decade, individuals and organizations have found it increasingly appealing to make use of remote services in the form of service-oriented architectures and cloud computing services. Data processed by remote services, however, is no longer under the direct control of the individual or organization that provided the data, leaving data owners at risk of data theft or misuse. This paper by F. Salim, N. P. Sheppard, and R. Safavi-Naini [3] describes a model called a rights management approach to securing data distribution in coalitions by which data owners can control the distribution and use of their data throughout a dynamic coalition of service providers using digital rights management technology. This model allows a data owner to establish the trustworthiness of every member of a coalition employed to process data, and to communicate a machine-enforceable usage policy to every such member.This implementation has serious threats to organizations.

Watermarking, which belong to the information hiding field, has seen a lot of research interest recently. There is a lot of work begin conducted in different branches in this field. Steganography is used for secret communication, whereas watermarking is used for content protection, copyright management, content authentication and tamper detection. In the paper of A Survey of Digital Image Watermarking Technique, V. M. Potdar, S. Han, and E. Chang [4] presented a detailed survey of existing and newly proposed steganographic and watermarking techniques. They classified the techniques based on different domains in which data is embedded. Here the survey has been limited to images only.

III Proposed Method

In the digital era, information leakage through unintentional exposures, or intentional sabotage by disgruntled employees and malicious external entities, present one of the most serious threats to organizations. It is not hard to believe that this is just the tip of the iceberg, as most cases of information leakage go unreported due to fear of loss of customer confidence or regulatory penalties. Large 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 leakage is very low, as there are currently almost no accountability mechanisms. For these reasons, the problem of data leakage has reached a new dimension nowadays.

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 proactively.

Propose different solutions; the former lets the sender open some pairs to validate that they are not equal and the latter uses oblivious transfer with a two-lock cryptosystem where the recipient can compare both versions in encrypted form. However, both proposed solutions have some flaws themselves. The problem is that it is possible to create two different versions with the same watermark, so even if the equality test fails, the two offered versions can still have the same watermark and the sender will know which watermark the recipient received. Also, the fix proposed in ruins the negligible probability of failure, as it does not split the document into parts, but creates n different. We see that all asymmetric fingerprinting protocols based on oblivious transfer that have been proposed so far suffer from the same weakness. We circumvent this problem in our

protocol by additionally sending a signed message including the watermark's content, so that the recipient is able to prove what he asked for. In contrast to the watermark, this message can be read by the recipient, so he can notice if the sender cheats.

The former lets the sender open some pairs to validate that they are not equal and the latter uses oblivious transfer with a two-lock cryptosystem where the recipient can compare both versions in encrypted form. However, both proposed solutions have some flaws themselves. The problem is that it is possible to create two different versions with the same watermark, so even if the equality test fails, the two offered versions can still have the same watermark and the sender will know which watermark the recipient received.

IV. INFORMATION LINEAGE FRAMEWORK LIME

In Micro benchmarking, we implemented the protocol in as a proof-of-concept and to analyze its performance. For the oblivious transfer sub protocol we implemented the protocol] using the PBC library, which itself makes use of the GMP library. For signatures we implemented the BLS scheme, also using the PBC library. For symmetric encryption we used an implementation of AES from the Crypto++ library. For watermarking we used an implementation of the Cox algorithm for robust image. We set the -factor, which determines the strength of the watermark, to a value of 0.1.

We executed the experiment with different parameters to analyze the performance. The sender and recipient part of the protocol are both executed in the same program, i.e., we do not analyze network sending, but only computational performance. The executing machine is a Lenovo ThinkPad model T430 with 8 GB RAM and 4 × 2.6GHz cores, but all executions were performed sequentially. We measured execution times for different phases of the protocol: watermarking, signature creation, encryption, oblivious transfer and detection. We executed each experiment 250 times and determined the average computation time and the standard deviation.

A) Possible Data Distortion:

We used a simple splitting algorithm: We split the image into n equally sized squares. However, when we used a strong watermark for the small parts (that is the - factor used by the Cox algorithm is 0.5), differences between adjacent parts became visible even though the single watermarks are imperceptible. This effect becomes even stronger after multiple iterations as observed. In some cases, this distortion might affect the usability of the document. We stress however, that we were still able to obtain good results with our approach. We used the Cox algorithm with an alpha factor of 0.1 and no distortion is visible. It might be interesting to investigate if this problem can be circumvented by using more elaborate splitting algorithms. As most watermarking schemes make use of the contiguity of information in the document, this is not a trivial task.

B) Broadcasting:

We present an approach for distributing data in a multicast system, so that every recipient holds a differently watermarked version. The sender splits the file into blocks and for each block he creates two different versions by watermarking them with different watermarks and encrypting them with different keys. Each recipient is assigned a set of keys, so that he can decrypt exactly one version of each part. The resulting combination of parts can uniquely identify the recipient. It showsanother approach for a broadcasting system that allows identification of recipients by their received files.

With a technique called finger casting, recipients automatically embed a watermark in files during the decryption process. The process is based on the chameleon cipher, which allows one to decrypt an encrypted file with different decryption keys, to introduce some noise that can be used as a means of identification. It use the technique of finger casting together with a randomized fingerprinting code in order to provide better security against colluding attackers. However, in these broadcasting approaches the problem of an untrusted sender is not addressed.

C) Watermarking:

Watermarking techniques have also been developed for other data types such as relational databases, text files and even Android apps. The first two are especially interesting, as they allow us to apply LIME to user databases or medical records. Watermarking [5] relational databases can be done in different ways. The most common solutions are to embed information in noise-tolerant attributes of the entries or to create fake database entries. For watermarking of texts, there are two main approaches. The first one embeds information by changing the text's appearance (e.g. changing distance between words and lines) in a way that is imperceptible to humans. The second approach is also referred to as language watermarking and works on the semantic level of the text rather than on its appearance.

A mechanism also has been proposed to insert watermarks to Android apps. This mechanism encodes a watermark in a permutation graph and hides the graph as a linked list in the application. Due to the list representation, watermarks are encoded in the execution state of the application rather than in its syntax, which

makes it robust against attacks. We propose an interesting approach for watermarking ontologies. In this approach the authors propose to rather remove existing information than adding new information or modifying existing information.

IV Experimental Results

In our project we are using jsp to design the process of an application. JSP pages easily combine static templates, including HTML or XML fragments, with code that generates dynamic content. JSP pages are compiled dynamically into Servlets when requested, so page authors can easily make updates to presentation code. JSP pages can also be precompiled if desired. Simulated screenshot for information lineage framework for LIME has been given in figure 1.

In our project we are using Servlets to control the application process. Servlets are modules that run within the server and receive and respond to the requests made by the client. Servlets retrieve most of the parameters using the input stream and send their responses using an output stream.

Figure 1:Local Host Data Lineage

The Java Collections API's provide Java developers with a set of classes and interfaces that makes it easier to handle collections of objects. In a sense Collection's works a bit like arrays, except their size can change dynamically, and they have more advanced behavior than arrays. In this project we are using Array List, Map and Set for saving values and do some function using that values. The information has to be submitted as shown in figure 2.

Figure 2: Submit details in data lineage

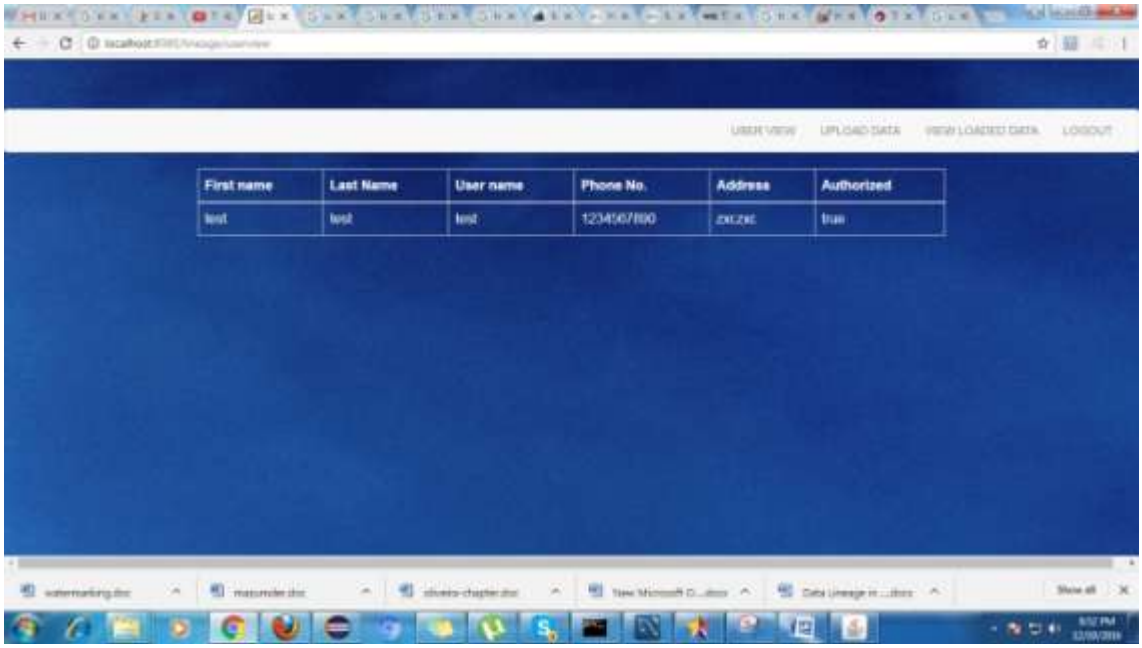

Figure 3: User View in Lineage Framework

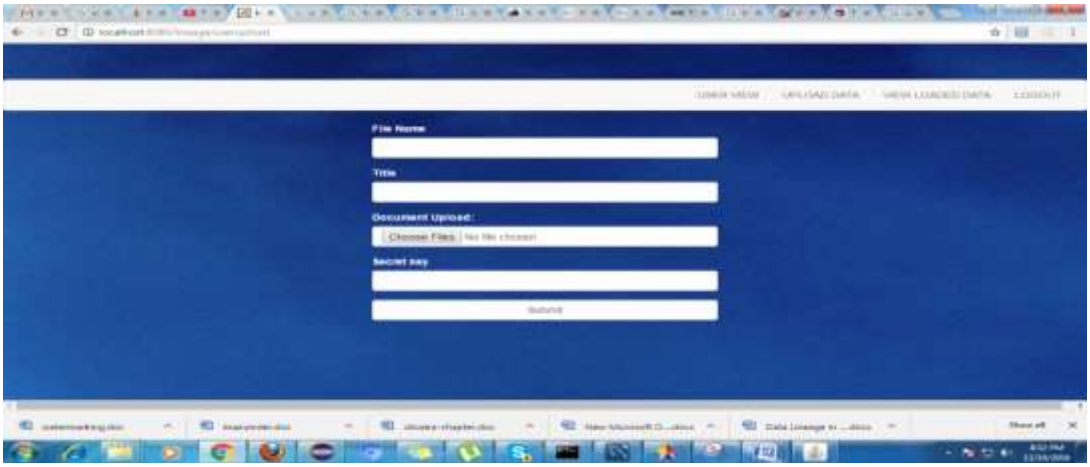

Figure 4: File upload

The Java Collections, Java API's provide Java developers with a set of classes and interfaces that makes it easier to handle collections of objects. Collections are useful to storing the values of a page as input to the backend and from backend to the front end carrying through network. Some of them are Array List, Hash Map.

In this project threading concept is very important. Why because, to increase the speed of the service process when more number of users interact. Thread is a independent part of the same program it may execute independently

Figure 5: Uploaded data

Figure 6: Consumer View

In our project we are using a backend as My SQL 5.5. Here we are create and maintaining the tables which are having values used for our processes. We are maintaining the registration table, update profile table, user updated profile table and shard file table. In our project we are using request process with the help of multi threading concepts.In our project we are using a backend as SQL Server 2005. Here we are create and maintaining the tables which are having values used for our processes. We are maintaining the registration table, login table and to store the values which is entered by the system admins and as well as end users

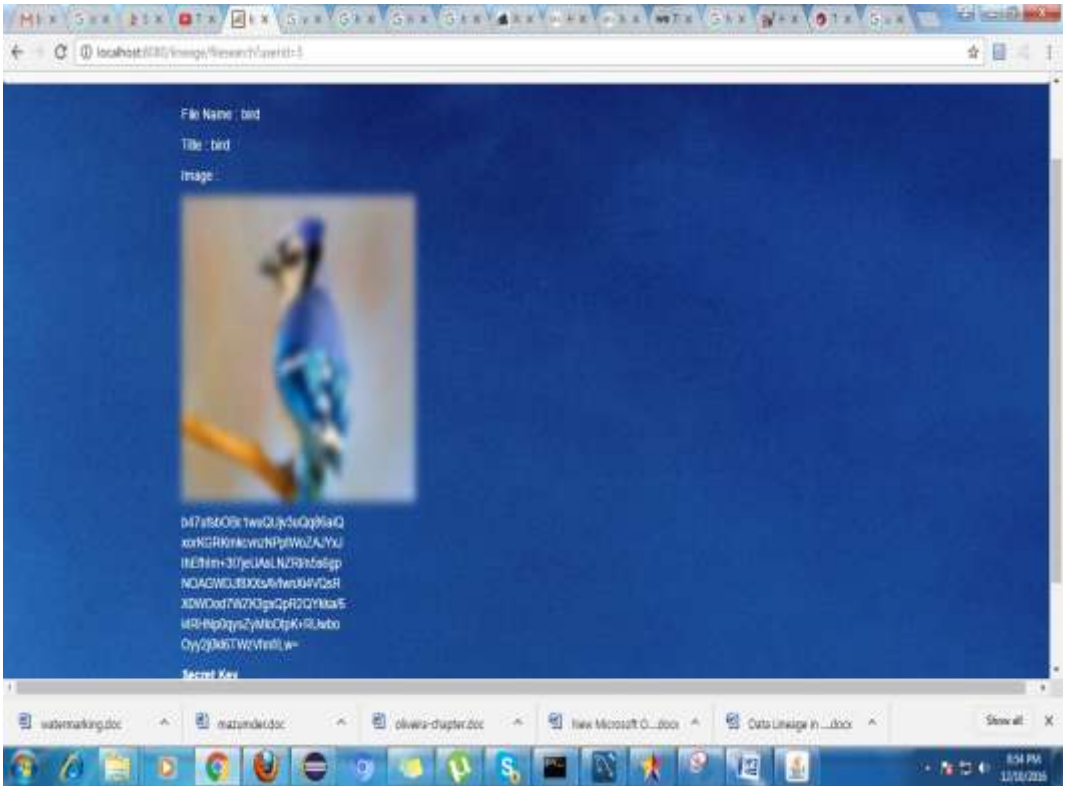

Figure 7: Local host Data Lineage framework

In this project threading concept is very important. A thread is a sequential path of code execution within a program. And each thread has its own local variables, program counter and lifetime. Like creation of a single thread, we can also create more than one thread (multithreads) in a program using class Thread or implementing interface runnable to make our project efficientand dynamic.

V Conclusion

In the project work carried out dynamic computation based LFSR and its decompresser design for its testing quality metrics and it is verified through modelsim based simulation which can produce pseudorandom test patterns with computation of seeds precisely. The switching activity can also be easily controlled by the pattern generator, so that the resultant test vectors can either yield a desired fault coverage faster than the conventional pseudorandom patterns while still reducing toggling rates down to desired levels, and offer visibly higher coverage numbers if run for comparable test times. To overcome the inefficiency arises to test deep registers and its counterpart of continuous switching from all register single cycle access methodology can be used. And this design is extended into fully functional test data decompressor with the ability to control scan shift-in switching activity through the process of encoding. The efficiency of proposed combine test compression with logic BIST is verified and proved to deliver high quality test.

References

[1]. Kelbert.F. and A. Pretschner, 2013, "Data usage control enforcement in distributed systems," in *CODASPY*, pp. 71–82.
[2]. Salim.F, N. P. Sheppard, and R. Safavi-Naini,2010, "A Rights Management Approach to Securing Data Distribution in Coalitions," in *NSS*, pp. 560–567.
[3]. M. J. Atallah, V. Raskin, C. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. E. Triezenberg, "Natural Language Watermarking and Tamperproofing",10.1007/3-540-36415-3_13, 2002.
[4]. V. M. Potdar, S. Han, and E. Chang, "A Survey of Digital Image Watermarking Technique",Industrial Informatics, 2005. INDIN '05. 2005 3rd IEEE International Conference 10.1109/INDIN.2005.1560462,2005
[5]. Adelsbach,A S. Katzenbeisser, and A.-R. Sadeghi,2007"A computational model for watermark robustness", in *Information Hiding*. Springer, pp. 145–160.