Secured Data Lineage Transmission in Malicious Environment

R. Gayathri[*1],K.Indumathi[*2],P. Githanjali[*3],V. Roobini[*4]
Department of Computer Science and Engineering
Sri Manakula Vinayagar engineering college, madagadipet, Pondicherry.

Abstract—**In this paper contains the fulfillment of Data Lineage in Malicious Environment. A data distributor has given precise data to a set of supposedly trusted agents. Some of the data are leaked and found in an unjustified place. The distributor must assess the likelihood that the crevice data came from one or more agents, as opposed to having been individually gathered by other means. We propose data allocation strategies that improve the probability of identifying crevices. These methods do not build on alterations of the released data. In some cases, we can also implant "realistic but fake" data records to further improve our chances of detecting crevice and identifying the guilty party. While sending data over the network there is lots of illegitimate user trying to get useful information. There should be proper security should be provided to data which is send to network. Now a days smartphones use have been increased rapidly and the applications used in smartphones can get easy access to our confidential information. So for avoiding this we used the data lineage mechanism. We give the fake information to guilty agent. We develop and analyze a novel accountable data transfer protocol between two entities within a malicious environment by building upon oblivious transfer, robust Watermarking, and signature primitives. Finally, we perform an experimental evaluation to demonstrate the practicality of our protocol and apply our framework to the important data leakage scenarios of data outsourcing and social networks. In general, we consider our lineage framework for data transfer, to be a key step towards achieving accountability by design.**

***Keywords*—Data storage system, Data leakage, Employing key, File assured deletion.**

I. Introduction

Data Leakage is an important concern for the business organizations in this increasingly networked world these days. Illegitimate disclosure may have serious consequences for an organization in both long term and short term. Risks include losing clients and stakeholder confidence, tarnishing of brand image, landing in undesirable lawsuits, and overall losing goodwill and market share in the industry. To prevent from all these unwanted and nasty activities from happening, an organized effort is needed to control the information flow inside and outside the organization. Here is our attempt to demystify the jargon surrounding the data leakage prevention procedures which will help you to choose and apply the best suitable option for your own business. Leakage describes an unwanted loss of something which escapes from its proper location and Lineage describes as data flow across multiple entities that take two characteristic, principal roles (i.e., owner and consumer). We define the exact security guarantees required by such a data lineage mechanism toward identification of a guilty entity, and identify the simplifying non-repudiation and honesty assumptions.

In the course of doing business, sometimes sensitive data must be handed over to supposedly trusted third parties. For example, a hospital may give patient records to researchers who will devise new treatments. Similarly, a company may have partnerships with other companies that require sharing customer data. Another enterprise may outsource its data processing, so data must be given to various other companies. The owner of the data can be called as distributor and the supposedly trusted third parties the agents. The goal is to detect when the distributors sensitive data have been leaked by agents, and if possible to identify the agent that crevice the data.

II. OVERVIEW OF DATA LINEAGE

Data Leakage Prevention is the category of solutions which help an organization to apply controls for preventing the unwanted accidental or malicious leakage of precise information to illegitimate entities in or outside the organization. Here sensitive information may refer to organization's internal process documents, strategic business plans, intellectual property, financial statements, security policies, network diagrams, blueprints etc.

2.1. Need Data Lineage

There are many fields where data leakage may occur, so it is very essential detect such
kind of detection, following users may lead to data leakage-
1. The security illiterate
• Majority of employees with little or no knowledge of security
• Corporate risk because of accidental breaches
2. The gadget needs
• Introduce a variety of devices to their work PCs
• Download software
3. The unlawful residents
• Use the company IT resources in ways they shouldn't
• i.e. by storing music, movies, or playing games
4. The malicious/disgruntled employees
• Typically minority of employees

• Gain access to areas of the IT system to which they shouldn't Send corporate data (e.g., customer lists, RD, etc.) to third parties.

2.2. Generic Data Leakage Prevention

- Deploy Security Mechanisms
- Firewalls, IDS's & antivirus software
- Thin-client architecture
- Advanced Security Measures
- Use of pattern based monitoring tools
- Use of reasoning algorithms
- Access Control & Encryption
- Access Control & Device Control
- Storage of encryption keys

III. RELATED WORK- CREATING ENCRYPTED DIGITAL WATERMARK

Our approach and watermarking are similar in the sense of providing agents with some kind of receiver identifying information. However, by its very nature, a watermark modifies the item being watermarked. If the object to be watermarked cannot be modified, then a watermark cannot be inserted. In such cases, methods that attach watermarks to the distributed data are not applicable. Finally, there are also lots of other works on mechanisms that allow only authorized users to access sensitive data through access control policies. Such approaches prevent in some sense data leakage by sharing information only with trusted parties. However, these policies are restrictive and may make it impossible to satisfy agents requests.

Fig: Encrypted Image using LIME

LIME (Lineage In the Malicious Environment) can be used with any type of data for which watermarking schemes exist. Therefore, we briefly describe different watermarking techniques for different data types. Most watermarking schemes are designed for multimedia files such as images, videos, and audio files. In these multimedia files, watermarks are usually embedded by using a transformed representation (e.g. discrete cosine, wavelet or Fourier transform) and modifying transform domain coefficients. 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 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. In this approach the authors propose to rather remove existing information than adding new information or modifying existing information. Thereby the watermarking scheme guarantees that no false entries are introduced. The above schemes can be employed in our framework to create data lineage for documents of the respective formats. The only modification that might be necessary when applying our scheme to a different document type is the splitting algorithm. For example for images it makes more sense to take small rectangles of the original image instead of simply taking the consecutive bytes from the pixel array. Embedding multiple watermarks into a single document has been discussed in literature and there are different techniques available. In they discuss multiple watermarking and in the focus is on segmented watermarking. Both papers show in experimental results that multiple watermarking is possible which is very important for our scheme, as it allows us to create a lineage over multiple levels. It would be desirable not to reveal the private watermarking key to the auditor during the auditor's investigation, so that it can be safely reused, but as discussed in current public key watermarking schemes are not secure and it is doubtful if it is possible to design one that is secure. In Sadeghi presents approaches to zero-knowledge watermark detection. With this technology it is possible to convince another party of the presence of a watermark in a document without giving any information about the detection key or the watermark itself. However, the schemes discussed in also hides the content of the watermark itself and are therefore unfit for our case, as the auditor has to know the watermark to identify the guilty person. Furthermore, using a technology like this would come with additional constraints for the chosen watermarking scheme.

IV. APPLICATION

It involves study of unobtrusive techniques for detecting leakage of a set of objects or records. Specifically, following scenario can be studied: After giving a set of objects to agents, the distributor discovers some of those same objects in an unauthorized place. At this point, the distributor can assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. In the proposed approach, a model is developed for assessing the guilt of agents. The algorithms are also presented for distributing objects to agents, in a way that improves the chances of identifying a leaker. Finally, the option of adding fake objects to the distributed set is also considered. Such objects do not correspond to real entities but appear realistic to the agents. In a sense, the fake objects act as a type of watermark for the entire set, without modifying any individual members. If it turns out that an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty. In the Proposed System, the hackers can be traced with good amount of evidence.

V. CONCLUSION & FUTURE SCOPE

We present LIME, a model for accountable data transfer across multiple entities. We define participating parties, their interrelationships and give a concrete instantiation for a data transfer protocol using a novel combination of oblivious transfer, robust watermarking and digital signatures.

Although LIME does not actively prevent data leakage, it introduces reactive accountability. Thus, it will deter malicious parties from leaking private documents and will encourage honest (but careless) parties to provide the required protection for sensitive data. LIME is flexible as we differentiate between trusted senders (usually owners) and untrusted senders (usually consumers). In the case of the trusted sender, a very simple protocol with little overhead is possible. The untrusted sender requires a more complicated protocol, but the results are not based on trust assumptions and therefore they should be able to convince a neutral entity (e.g.a judge). Our work also motivates further research on data leakage detection techniques for various document types and scenarios. For example, it will be an interesting future research direction to design a verifiable lineage protocol for derived data.

REFERENCES

[1] M. ARMBRUST, A. FOX, R. GRIFFITH, A.D. JOSEPH, R. KTAZ, A. KONWIN-SKI, G. LEE, D. PATTERSON, A. RABKIN, I. STOICS, AND M. ZAHARIA, “A VIEW OF CLOUD COMPUTING,” *COMMUNICATIONS OF THE ACM*, VOL. 53, NO. 4, 2010, PP. 50-58.

[2] M. S. BLUMENTHAL, “IS SECURITY LOST IN THE CLOUDS?”*COMMUNICA-TIONS AND STRATEGIES*, NO. 81, 2011, PP. 69-86.

[3] C.CACHINAND M.SCHUNTER, "A CLOUD YOU CAN TRUST,"*IEEESPECTRUM*,VOL. 48, NO. 12,2011,PP. 28-51.

[4] C. CREMERS, "THE SCYTHER TOOL: VERIFICATION, FALSIFICATION, AND ANALYSIS OF SECURITY PROTOCOLS." *IN COMPUTER AIDED VERIFICATION, SPRINGER BERLIN HEIDELBERG*, 2008, PP. 414-418.

[5]CLOUD SECURITY ALLIANCE HTTPS://DOWNLOADS.CLOUDSECURITYALLIANCE.ORG/INITIATIVES/CDG/ CSA_CCAQIS_SURVEY.PDF (ACCESSED MARCH 24, 2013).

[6] W. DIFFIE, P. C. V. OORSCHOT, AND M. J. WIENER, “AUTHENTICATION AND AUTHENTICATED KEY EXCHANGES,”*DESIGNS, CODES AND CRYPTOGRAPHY*,VOL. 2, NO. 2, 1992, PP. 107-125.

[7]M. Ali, K. Bilal, S. U. Khan, B. Veeravalli, K. Li, and A. Y. Zomaya, “DROPS: Division and Replication of Data in the Cloud for Opti-mal Performance and Security,” *IEEE Transactions on Cloud Computing*, 2015, DOI: 10.1109/TCC.2015.2400460.

[8] N. En and N. Srensson, “An extensible SAT-solver,” *Lecture Notes in Computer Science*, vol. 2919, Springer, 2003, pp. 502-518.

[9] C P. Gomes, H. Kautz, A. Sabharwal, and B. Selman, “Satisfia-bility solvers,” *In Handbook of Knowledge Representation, Elsevier*, 2007.

[10] M. Ali, S. U. Khan, and A. V. Vasilakos, “Security in cloud compu-ting: Opportunities and challenges,”*Information Sciences*,Vol. 305, 2015, pp. 357-383.

[11] A. Juels and A. Opera, “New approaches to security and availabil-ity for cloud data,” *Communications of the ACM*,Vol. 56, No. 2, 2013, pp. 64-73.

[12] S. Kamara and K. Lauter, “Cryptographic cloud storage,” *Financial Cryptography and Data Security*,Springer Berlin Heidelberg, 2010, pp. 136-149.

[13]M. Kaufman,”Data security in the world of cloud compu-ting,”*IEEE Security andPrivacy*,Vol. 7, No. 4, 2009, pp. 61-64.

[14] H. Lin and W. Tzeng, “A secure decentralized erasure code for distributed network storage,” *IEEE Transactions on Parallel and Dis-tributed Systems*, vol. 21, no. 11, Nov. 2010, pp. 1586-1594.

[15] H. Lin and W. Tzeng, “A secure erasure code-based cloud storage system with secure data forwarding,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 6, June 2012, pp. 995-1003.

[16] S. U. R. Malik, S. K. Srinivasan, S. U. Khan, and L. Wang, “A Methodology for OSPF Routing Protocol Verification,” *12th Inter-national Conference on Scalable Computing and Communications (ScalCom)*, Changzhou, China, Dec. 2012.

[17] L. Moura and N. Bjrner, “Satisfiability Modulo Theories: An appetizer,” *Lecture Notes in Computer Science*, Vol. 5902, Springer, 2009, pp. 23-36.

[18] T. Murata, “Petri Nets: Properties, Analysis and Applications,” *Proc. IEEE*, Vol. 77, No. 4, pp. 541-580, Apr. 1989.

[19] A. Shamir, “How to Share a Secret,” *Comm. ACM*, Vol. 22, No. 11, Nov. 1979, pp. 612-613.

[20]H. Takabi, J. B. D. Joshi, and G. J. Ahn, "Security and privacy challenges in cloud computing environments,"*IEEE Security andPrivacy*, Vol. 8, No. 6, 2010,pp. 24-31.

[21] Y. Tang, P. P. Lee, J. C. S. Lui, and R. Perlman, “Secure Overlay Cloud Storage with Access Control and Assured Deletion,” *IEEE Transactions on Dependable and Secure Computing*, Vol. 9, No. 6, Nov. 2012, pp. 903-916.

[22]A. Yun, C. Shi, and Y. Kim, “On protecting integrity and confiden-tiality of cryptographic file system for outscored storage,” *Proceed-ings of 2009 ACM workshop on cloud computing security CCSA’09*, pp. 67-76, 2009.

[23] M. Ali, R. Dhamotharan, E. Khan, S. U. Khan, A. V. Vasilakos, K. Li, and A. Y. Zomaya, “SeDaSC: Secure Data Sharing in Clouds,” *IEEE Systems Journal*,2015,http:// dx.doi.org/10.1109/JSYST.2014.2379646.

[24] A. R.Khan, M. Othman, S. A. Madani, and S. U. Khan, “A survey of mobile cloud computing application models,”*IEEE Communica-tions Surveys and Tutorials*,2013,1-21.