DATA CONTAINS LEAK PROTECTION AND ANALYSIS

[1] Swati P. Bisen ,[2] Kapesh Raghatate

[1] Student CSE, RCERT Chandrapur [2] Professor CSE, RCERT Chandrapur,

Abstract : The information leak of sensitive data on systems has a serious threat to organization data security. Statistics show that the improper encryption on files and communications due to human errors is one of the leading causes of information loss. However, detecting the exposure of sensitive data information is challenging due to data transformation in the content. Transformations result in highly unpredictable leak patterns. In this paper, it is utilize sequence alignment techniques used for detecting complex data-leak patterns. This algorithm is designed for detecting long and inexact sensitive data patterns. The system achieves good detection accuracy in recognizing transformed leaks. To demonstrate the high multithreading scalability of the data leak detection method required by a requirement of organization.

IndexTerms **-** Information leak detection, content inspection, sampling, alignment, dynamic programming, etc..

I. INTRODUCTION

To sensitive data and documents minimize the exposure of, an company needs to prevent text sensitive data from appearing in the storage server. In increasingly digital world, there is often a tension between safeguarding privacy and sharing information. Although, in general, sensitive data clearly needs to be kept confidential data owners are often motivated, or forced, to share sensitive information Privacy-Preserving Sharing of Sensitive Information , and proposes one efficient and secure instantiation that functions as a privacy shield to protect parties from disclosing more than the required minimum of sensitive information. We model in the context of simple database-querying applications with two parties: a server that has a database, and a client, performing simple disjunctive equality queries Detecting the exposure of sensitive information is challenging due to data transformation in the content. Transformations result in highly unpredictable leak patterns. In this paper, sequence alignment the utilize techniques for data-leak detecting complex asymmetric cryptography, facilitate the creation of a verifiable association between a public key and the identity other attributes of the holder of the corresponding private key, for uses such as authenticating the identity of a specific entity, ensuring the integrity of information, providing support for non repudiation, and establishing an encrypted communications section. Cloud Computing-Based Forensic Analysis for Collaborative Network Security Management System: Zhen Chen*, Fuye Han, Junwei Cao, Xin Jiang, and Shuo Chen.1, February 2013. The problems remain a major challenge with many security Internet security concerns such as Internet worms, spam, and phishing attacks. Botnets, distributed network attacks, of a large number of bots that generate big volumes of spam or launch Distributed Denial of Service (DDoS) attacks on victim hosts. A overlay network distributed security with a centralized security center leverages a peer-to-peer communication protocol used in the UTMs collaborative module. These new security rules are enforced by collaborative UTM and the feedback events of such rules are returned to the security center. Collaborative network security management system can not identify the intrustion.[6] Understanding privacy in data mining requires understanding how privacy can be violated and the possible means for preventing privacy violation. In general, one major factor contributes to privacy violation in data mining: the misuse of data. The privacy different ways can be violated in and with different intentions. Although data mining can be extremely valuable in many applications it can also, in the absence of adequate safeguards, violate informational privacy.

II. RESEARCH METHOD

2,1 Identity Key Generation The key generation module helps the users to share the information between source and destination. After getting the confirmation response from the receiver side the sender fix the information and encrypt it. At this time a key will be generated and sent to the receiver area. That key is useful for decrypt the data at receiver end. As well as an entity that stores data from senders and provide corresponding access to users. It may be mobile or static. Similar to the previous schemes, and also assume the storage node to be semi trusted that is honest but curious.

2,2 3DES Based Encryption In Ciphertext Policy Attribute based Encryption scheme, the encryptor can fix the policy, who can decrypt the encrypted message. The policy can be formed with the help of attributes. In access policy is sent along with the ciphertext. We propose a method in which the access policy need not be sent along with the ciphertext, by which we are able to preserve the privacy of the encryptor. This techniques encrypted data can be kept confidential even if the storage server is untrusted. Moreover, our methods are secure against collusion attacks. Previous Attribute Based Encryption systems used attributes to describe the encrypted data and built policies into user's keys; while in our system attributes are used to describe a user's credentials, and a party encrypting data determines a policy for who can decrypt.

2,3 Confidential Data Interchange This is an entity who owns confidential messages or data and wishes to store them into the external data storage node for ease of sharing or for reliable delivery to users in the extreme networking environments. A sender is responsible for defining (attribute based) access policy and enforcing it on its own data by encrypting the data under the policy before storing it to the storage node. This is a mobile node who wants to access the data stored at the storage node (e.g., a soldier). If a user possesses a set of attributes satisfying the access policy of theencrypted data defined by the sender, and is not revoked in any of the attributes, then he will be able to decrypt the ciphertext and obtain the data. Administrative Access

Controller The administrator owns full access rights of this entire site. Once the administrator find out any illegal activity or other misusing happens into the way of transaction between the respective sender and receiver then the admin immediately block the user access rights to transact using this site. The block will be unblocked after getting meaningful reason from the user end.

2.4 THE .NET FRAMEWORK

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).
2. A hierarchical set of class libraries.

The CLR is described as the "execution engine" of .NET. It provides the environment within which programs run. The most important features are

- Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
- Memory management, notably including garbage collection.
- Checking and enforcing security restrictions on the running code.
- Loading and executing programs, with version control and other such features.
- The following features of the .NET framework are also worth description:

2.5 Managed Code

The code that targets .NET, and which contains certain extra Information - "metadata" - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

2.6 Managed Data

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you're using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn't get garbage collected but instead is looked after by unmanaged code.

Flow chart

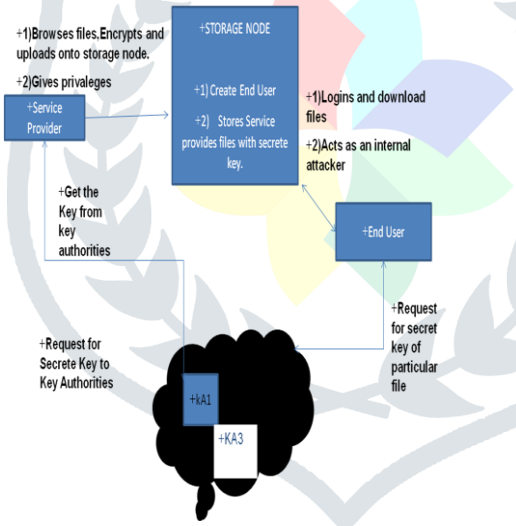
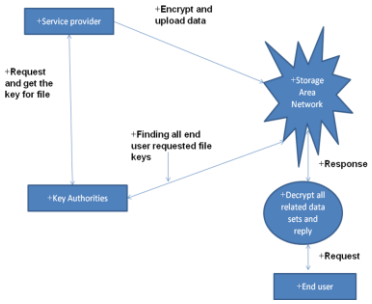


Figure 1 Dataflow diagram

The A typical setting involves two parties: one that seeks information from the other that is either motivated, or compelled, to share (only) the requested information. Consequently, in numerous occasions, there is a tension between information sharing and privacy. On the one hand, sensitive data needs to be kept confidential; on the other hand, data owners may be willing, or forced, to share information. We extensively evaluate the accuracy of our solution with several types of datasets under a multitude of real-world data leak scenarios.

III. IDENTITY AUTHORIZATION

This module allows the user to register their identity into the system with proper input parameters. The key generation centers play a vital role in it, which generates public/ secret parameters . The key authorities consist of a central authority and multiple local authorities. Assume that there are secure and reliable communication channels between a central authority and each local authority during the initial key setup and generation phase. Each local authority manages different attributes and issues corresponding attribute keys to users. They grant differential access rights to individual users based on the users' attributes. The key authorities are assumed to be honest but curious. That is, they will honestly execute the assigned tasks in the system however they would like to learn information of encrypted contents as much as possible.



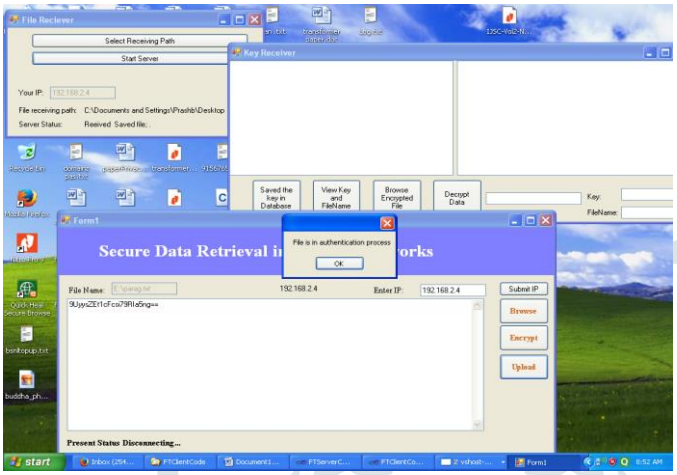


Figure 2 Data Verification system

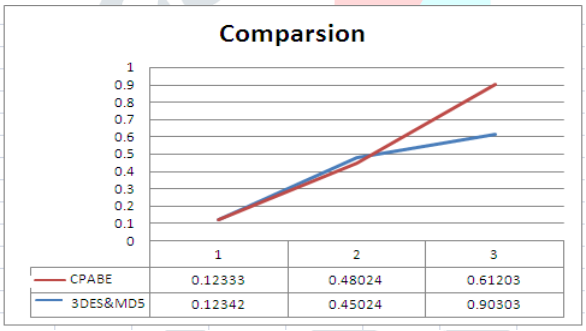


Figure 3 Result Comparison

In my paper security is combination of more algorithm than base paper still requires less time to

Verify and process.
These are not present in the base paper in my project to enhance the security we use combination of algorithm.

1. Idea algorithm
2. MD5
3. ECB (ELECTRONIC CODE BOOK)
4. Hashing code

VI. Confidentiality :

In order to protect sensed data and communication ex-changes between sensor nodes it is important to guarantee the secrecy of messages. In the sensor network case this is usually achieved by the use of symmetric cryptography as asymmetric or public key cryptography in general is considered too expensive. However, while encryption protects against outside attacks, it does not protect against inside attacks/node compromises, as an attacker can use recovered cryptographic key material to successfully eavesdrop, impersonate or participate in the secret communications of the network. Furthermore, while confidentiality guarantees the security of communications inside the network it does not prevent the misuse of information reaching the base station. Hence, confidentiality must also be coupled with the right control policies so that only authorized users can have access to confidential information.

Integrity and Authentication Integrity and authentication is necessary to enable sensor nodes to detect modified, injected, or replayed packets. While it is clear that safety-critical applications require authentication, it is still wise to use it even for the rest of applications since otherwise the owner of the sensor network may get the wrong picture of the sensed world thus making

inappropriate decisions. However, authentication alone does not solve the problem of node takeovers as compromised nodes can still authenticate themselves to the network. Hence authentication mechanisms should be "collective" and aim at securing the entire network.

In particular, the following requirements must be supported by the key management scheme, in order to facilitate data aggregation and dissemination process:

1. Data aggregation is possible only if intermediate nodes have access to encrypted data so that they can extract measurement values and apply to them aggregation functions. Therefore, nodes that send data packets toward the base station must encrypt them with keys available to the aggregator nodes.
2. Data dissemination implies broadcasting of a message from the aggregator to its group members. If an aggregator shares a different key (or set of keys) with each of the sensor within its group, then it will have to make multiple transmissions, encrypted each time with a different key, in order to broadcast a message to all of the nodes. But transmissions must be kept as low as possible because of their high energy consumption rate.

V. Conclusion

Detecting multiple common data leak scenarios. The parallel versions of our prototype provide substantial speedup and indicate high scalability of our design. For future work, we plan to explore data-movement tracking approaches for data leak prevention on a host. Privacy guarantees are formally defined and achieved with provable security. Experimental results show that our approach is sufficiently efficient for real-world applications. To support efficient handling of multiple auditing tasks, we further explore the technique of bilinear aggregate signature to extend our main result into a multi-user setting, where can perform multiple auditing tasks simultaneously. Extensive security and performance analysis shows the proposed schemes are provably secure and highly efficient.

REFERENCES

[1] Hiroki Nishiyama, Desmond Fomo, Zubair Md. Fadlullah,, and NeiKato,Fellow," Traffic Pattern Based Content Leakage Detection for Trusted Cont ent Delivery Networks" IEEE Transaction on Parallel and Distributed System , Volume 25, No 2 Feb 2014

[2] K. Ramya, D. RamyaDorai, Dr. M. Rajaram "Tracing Illegal Redistributors of Streaming Contents using Traffic Patterns" IJC A 2011

[3] A. Asano, H. Nishiyama, and N. Kato, "The Effect of Packet Reordering and Encrypted Traffic on Streaming Content Leakage Detection" Proc. Int'l Conf. Computer Comm. Networks (ICCCN '10), pp. 1 6, Aug. 2010.

[4] Y. Chu, S.G. Rao, S. Seshan, and H. Zhang, "Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture," Proc.ACM SIGCOMM, pp. 55 67,Aug. 2010

[5] O. Adeyinka, "Analysis of IPSec VPNs Performance in a Multimedia Environment," Proc. Fourth Int'l Conf. Intell igent Environments, pp. 25 - 30, 2008

[6] M. Dobashi, H. Nakayama, N. Kato, Y. Nemoto, and A. Jamalipour, "Traitor Tracing Technology of Streaming Contents Delivery Using Traffic Pattern in Wired/Wireless Environments," Proc. IEEE Global Telecomm. Conf., pp. 1 5, Nov./Dec. 2006.

[7] S. Amarasing and M. Lertwatechakul, "The Study of Streaming Traffic Behavior," KKU Eng. J., vol. 33, no. 5, pp. 541 553, Sept./Oct. 2006.

[8] R.S. Naini and Y. Wang, "Sequential Traitor Tracing," IEEE Trans. Information Theory, vol. 49, no. 5, pp. 1319 1326, May 2003.

[9] D. Geiger, A. Gupta, L.A. Costa, and J. Vlontzos, "Dynamic Programming for Detecting, Tracking, and Matching Deformable C ontours," Proc. IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 17, no. 3, pp. 294 302, M ar. 1995.