ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

Encrypted Query Processing Based Log Management of the Cloud for Improved Protection of Confidentiality

Sinu P S [1], M.Ananthi[2]

[1] PG Scholar, Info Institute Of Engineering, India.

[2] Assistant Professor, Department of CSE., Info Institute of Engineering., India,

ABSTRACT: Logging is an important part in central service computing systems. It provides the foundation for accountability and audit services in systems and for other accessory services While logging services are provided a traditional computing systems is relatively smooth process, where it turns to be an intricate process. A log record contains the track of events which was occurring within an organization's systems and networks. Logs include log entries, where each entry contains information that are related to a specific event that had occurred within systems . Logs are primarily used for troubleshooting problems, but logs contain many functions within most organizations, such as optimizing system and network performance, recording the actions of users, and providing data useful for investigating malicious activity. The privacy of the log is concerned with the current implementation which allows access to log records that are indirectly identified by upload-tag values. The proposed system includes a practical homomorphic encryption schemes that will allow encryption of log records in such a way that the logging cloud can execute some queries on the encrypted logs without breaching confidentiality or privacy. It reduces the communication overhead between a log monitor and the logging cloud to the most such that to the answer required to the queries on logs.

KEYWORDS: Cloud computing, logging, privacy, security.

I. INTRODUCTION

Cloud engineering [6] is the application of engineering disciplines to cloud computing. It brings a systematic approach to the high level concerns of commercialization, standardization, and governance in conceiving, developing, operating and maintaining cloud computing systems. It is a multidisciplinary method encompassing contributions from diverse areas such as systems, software, web, performance, information, security, platform, risk, and quality engineering.

A number of approaches have been proposed for logging. The approaches that have been proposed are based on syslog based environment which is a standard for network based logging protocol. The protocol used by syslog is UDP for transferring log information from client to the log server. Therefore delivery of log messages is not reliable.

The disadvantage of syslog is that it does not protect log records during transmit or at any point-point transfer. Syslog-ng is the replacement for backward compatible with syslog. Some of the features include supporting IPv6 protocol, capability to transfer log messages with reliability using TCP, and filtering the content of logs using regular expressions. Syslog-ng describes the encryption of log record using SSL during the transmission such that it protects the data from confidentiality and integrity precluding during the transmit. Syslog-ng does not protect log data modifications when it is placed at an end-point.

The next concept is Syslog-sign which is an enhanced model of syslog that includes the addition of authentication, message integrity, replay resistance, message sequencing, and detection of missing messages by using two additional messages —"signature blocks" and "certificate blocks."

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

But, if signature blocks associated with the log records get deleted after authentication, which tampers the evidence and forward integrity is only partially fulfilled. Syslog-sign also do not provide any confidentiality or privacy during the transmission of data or at any end-end transmission.

The another concept related to syslog is Syslog-pseudo which is also an enhancement of syslog that proposes a logging architecture for pseudonymizing log files. The main idea behind this is: the log records are first processed by the pseudonymizer (before being archived), therefore the pseudonymizer filters out identifying the features from the specific fields that are in the log record and substitutes them with the ones that are carefully crafted with pseudonyms. Therefore, this protocol do not ensure the correctness of the logs

Neither syslog-pseudo nor anonymous log file anonymizer protect the log records from the attacks of security i.e., violations of confidentiality and integrity and the other end-point attacks. The main aim of Reliable-syslog is to implement reliable delivery of syslog messages, which is being built on the top of the blocks of extensible exchange protocol (BEEP) that runs over TCP for providing the required reliable delivery service. The Reliable-syslog protocol has been included with the device authentication and the incorporation of mechanisms for the protection of integrity of log messages and also to protect the log messages against replay attacks and also from the other attacks that might affect the log data.

The protocol for a forward-integrity log record was proposed by Bellare and Yee which was mainly developed for protecting the pre compromising of log data from post compromising insertion, deletion, modification, and reordering. Forward integrity is established by the usage of secret key mechanism that became the starting point of a hash-chain. The hash-chain is generated cryptographically by strong one-way function where the key is being changed for each and every log record.

Schneier and Kelsey proposed scheme as :

A logging scheme for cloud that depends on the forward integrity and provides assurance related to it. This scheme is based upon the forward-secure message authentication codes and one-way hash chaining which is moreover similar to the suggestion by the Bellare-Yee protocol i.e., if the trusted server is being attacked or compromised, it breaks the security of the logging scheme.

Holt has improved the Schneier-Kelsey protocol by merging all the public verifiability log records. This process is highly based on public-key scheme, the overhead found here is significantly high.

Therefore none of the above specified three schemes consider the privacy of storing and retrieving log records.

Additionally all these three schemes suffer equally from truncation attacks.

II. RELATED WORK

The following have been analysed and studied in order to Manage the Log:.

K. Kent and M. Souppaya [1] proposed as: Log management is benefited in an organization in many ways. It helps in ensuring the computer security records that are stored in sufficient detail for an appropriate period of time. Routine log reviews and analysis are beneficial for identifying the following factors: security incidents, policy violations, fraudulent activity, and operational problems which are shortly carried out after they have occurred, and for providing information useful for resolving such problems. Log management is driven by reasons of security,[2] system and network operations (such as system or network administration) and regulatory compliance. Effectively analyzing large volumes of diverse logs can pose many challenges — such as huge log-volumes (reaching hundreds of gigabytes of data per day for a large organization), log-format diversity, undocumented proprietary log-formats (that resist analysis) as well as the presence of false log records in some types of logs (such as intrusion-detection logs).

D. New and M. Rose [2] stated in their work as: The BSD Syslog Protocol prescribes a number of service related options therefore to relate the propagating event messages. In telecommunications and computing, a product or technology is backward or downward compatible if it can work with input generated by an older product or technology.[11] If products designed for the new standard can receive, read, view or play older standards or formats,

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

then the product is said to be backward-compatible; examples of such a standard include data formats and communication protocols. Modifications to a system that do not allow backward compatibility are sometimes called "breaking changes."

M. Bellare and B. S. Yee [3] et al discussed as: Applications include securing audit logs (e.g., syslogd data) for intrusion detection or accountability, communications security, and authenticating partial results of computation for mobile agents. Computer audit logs contain descriptions of noteworthy events which crashes of system programs, system resource exhaustion, failed login attempts, etc. To make the audit log secure, it is a responsibility to prevent the attacker from modifying the audit log data.

J. E. Holt [4] projected as: The popular application Tripwire keeps cryptographic fingerprints of all files on a computer allowing administrators to detect when the attackers compromise the system and modify the important system files. Tripwire delivers unprecedented risk visibility, business context and security business intelligence enabling enterprises to protect sensitive data and assets from breaches, vulnerabilities, and threats, through our trusted portfolio of high priority security controls including:

- Security Configuration Management
- Vulnerability Management
- File Integrity Monitoring
- Log & Event Management
-

B. Schneier and J. Kelsey [5] concentrated on the following: In real-world applications, sensitive information are kept in logs which are on an untrusted machine. When an event occur as an attacker captures this machine, it is guaranteed that the attacker will gain little or no information from the log less and to limit his ability to corrupt the log files. Here the proposed system describes a cheap method for IDS come in a variety of “flavors” and approach the goal of detecting suspicious traffic in different ways. There are network based (NIDS) and host based (HIDS) intrusion detection systems. Some systems may attempt to stop an intrusion attempt but this is neither required nor expected of a monitoring system. Intrusion detection and prevention systems (IDPS) are primarily focused on identifying possible incidents, logging information about them, and reporting attempts. In addition, organizations use IDPSes for other purposes, such as identifying problems with security policies, documenting existing threats and deterring individuals from violating security policies. IDPSes have become a necessary addition to the security infrastructure of nearly every organization.[17]

D. Ma and G. Tsudik [6] discussed as : The need for secure logging is well-understood by the security professionals, including both researchers and practitioners. The ability to efficiently verify all log entries is more important for any application to employ secure logging techniques. In this paper, by examining the state-of-the-art in secure logging and by the identification of some problems by inherenting systems based on trusted third-party servers. They propose a different approach for secure logging based upon th recently developed Forward-Secure Sequential Aggregate (FssAgg) authentication techniques.

D. Dolev and A. Yao [7] et al proposed as: Recently the use of public key encryption was to provide a secure network communication which has received a considerable attention. Public-key algorithms are based on mathematical problems which currently admit no efficient solution that are inherent in certain integer factorization, discrete logarithm, and elliptic curve relationships. It is computationally easy for a user to generate their own public and private key-pair and to use them for encryption and decryption. The strength lies in the fact that it is "impossible" (computationally unfeasible) for a properly generated private key to be determined from its corresponding public key. Thus the public key may be published without compromising security, whereas the private key must not be revealed to anyone not authorized to read messages or perform digital signatures. Public key algorithms, unlike symmetric key algorithms, do *not* require a secure initial exchange of one (or more) secret keys between the parties.

G. R. Blakley [8] suggested as: Certain cryptographic key, involves a number of components that makes possible components to compute the secret decoding exponents in a RSA public key cryptosystem 1,5 or the system master key and certain other keys in a DES cryptosystem , 3 are important as they present a dilemma.

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

There are two principles for counting incidents.

- The first incident is Boole's law of inclusion and exclusion.

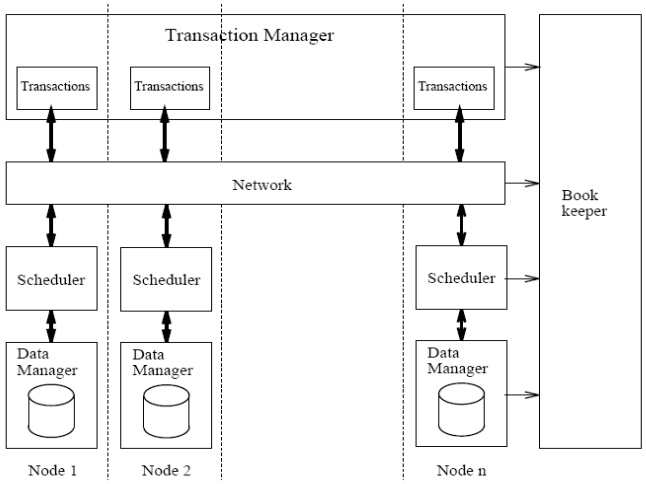
Consider that an organization issues are non-volatile; the pieces of information guards and waits with a modest period of time during which incidents can occasionally occur.

Let *a* stand for the number of abnegation incidents, *b* for the number of betrayal incidents and *c* for the number of combination incidents. The total number *d* of incidents is $d=a+b-c$ because a combination incident gets counted twice, once by *a* and once by *b*.

- The second principle is that the incidents are so rare, such that the possibility of two separate incidents occurring with the same non-volatile piece of information is usually dismissed on probabilistic grounds as absurd within the range.

A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung [9] discussed the following: Secret sharing schemes were identified to protect the secrets by distributing them over different locations. In particular k out of n threshold schemes, security is being assured for its entire life time is ought to be secret and therefore its adversary is restricted to compromise less than k of the n location for long lived and sensitive secret protection is insufficient. Hence the proposed system seem to be an efficient proactive secret sharing scheme, here the shares are periodically renewed such that the information gained by the adversary in one period of time is useless for the other.

K. Nørvag, O. Sandsta [10] et al concentrated on simulating distributed database systems which is much difficult to systemize, where there are many factors which may influence the resultant part. This paper includes architectural options as well as distribution of workload and data. They also present a DBsim simulator and some simulated results. The DBsim simulator architecture is easily extendible, and is easy for changing the parameters and the configuration. The simulated results are the comparison of both performance and response times for different algorithms. The simulations have been done by running it in more number of nodes, network types, data de-clustering and workloads. The results are shown in a mix of small and long transactions, and the throughput that is generated is significantly higher for a system when it is compared with the timestamp ordering scheduler than for a system with a two-phase locking scheduler. The short transactions are only for, the performance of the two schedulers which are almost identical when compared with results. The Long transactions are treated fairly by a two-phase locking scheduler, i.e., timestamp ordering scheduler has higher abort rate for long transactions.



2.1 The architecture of the simulator.

The above model is a model for simulator of distributed page-server based object-oriented database system. This simulator is modular and extendible. In future work, the DBsim simulator include extensions that would make the

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

simulator more suitable for simulation of algorithms for object-oriented databases. Therefore, much can be done with both simulation model and with the simulator. This includes adding new schedulers to the system. In real system, replication is used for increased reliability and performance. This can also be coupled with this framework.

III. MATERIALS AND METHODS

The Log Generators, Logging Client or Logging Relay, Logging Cloud and Log Monitor. Create those file and connect them to form a cloud.

The protocol starts with the logging client randomly generating three master keys — A_0 and X_0 for ensuring log integrity, and K_0 for ensuring log confidentiality. These keys should satisfy the requirements of the chosen proactive secret-sharing scheme. It is then embarked upon preparing the log records. Log data is arriveed at the logging client as a series of messages $L_1, L_2, \ldots, L_n$. Each L_i contains a group of log records generated by a log generator. Assume that these messages are transmitted to the logging client over an authenticated network. The logging client uploads is a preparation of log records with batches of n. The value n is determined randomly at the beginning of each log batch preparation.

1. Before any log data arrives at the logging client, the logging client creates a special first log entry L_0 = {TS, log-Initialization, n}. It then encrypts this log entry with the key K_0 and computes the message authentication code $MAC_0 = H_{A0}\ [E_{K0}[L_0]]$ for the encrypted entry with the key A_0. The client adds the resulting first log entry for the current batch—$\{E_{K0}[L_0], MAC_0\}$—to the log file.
2. The logging client then computes new set of keys $A_1 = H[A_0]$, $X_1 = H[X_0]$ and $K_1 = H[K_0]$, securely erases the previous set of keys and waits for the next log message to arrive.
3. When the first log message L_1 arrives, the logging client creates a record $M_1 = L_1 \| H_{A0}\ [E_{K0}\ [L_0]$. It encrypts M_1 with the key K_1, and creates a message authentication code for the resulting data as $MAC_1 = H_{A1}\ [E_{K1}[M_1]]$. It also computes an aggregated message authentication code $MAC'_1 = H^n_{X1}[MAC_0 \| MAC_1 \| n]$. The log batch entry is $\{E_{K1}\ [M_1], MAC_1\}$. It then creates the next set of keys $A_2 = H[A_1]$, $X_2 = H[X_1]$ and $K_2 = H[A_1]$ and securely deletes A_1 and K_1.
4. For every new log data L_i that the logging client subsequently receives, it creates log file entries $\{E_{Ki}\ [M_i], MAC_i\}$, where $M_i = L_i \| MAC_{i-1}$ and $MAC_i = H_{Ai}\ [E_{Ki}\ [M_i]]$. It also creates the aggregated message authentication code $MAC'_i = H^{n-i+1}{}_{Xi}\ [MAC_{i-1} \| MAC_i] \| n - i + 1$. Once MAC'_i has been generated, MAC^{i-1} is securely deleted. The client finally creates new keys $A_{i+1} = H[A_i]$, $X_{i+1} = H[X_i]$ and $K_{i+1} = H[K_i]$ and securely erases the keys A_i, X_i and K_i.
5. After the client creates the last log entry M_n for the current batch from the last log data Ln, it creates a special log close entry $LC = \{E_{Kn+1}[TS, log\text{-}close \| MAC_n], H_{An+1}\ [E_{Kn+1}[TS, log\text{-}close \| MAC_n]]\}$, and an aggregated message authentication code MAC'_{n-1}. It then securely erases the three keys used in this step and uploads the resulting log batch and aggregated message authentication code to the logging cloud as one unit.

Tor is free software for enabling online anonymity. Tor directs Internet traffic through a free, worldwide volunteer network consisting of more than three thousand relays to conceal a user's location or usage from anyone conducting network surveillance or traffic analysis. Using Tor makes it more difficult to trace Internet activity, including "visits to Web sites, online posts, instant messages and other communication forms", back to the user and is intended to protect users' personal privacy, freedom, and ability to conduct confidential business by keeping their internet activities from being monitored.

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

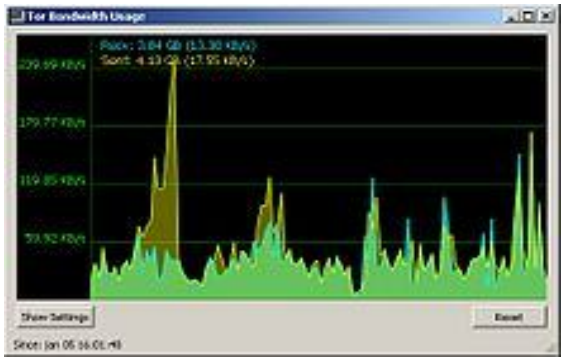
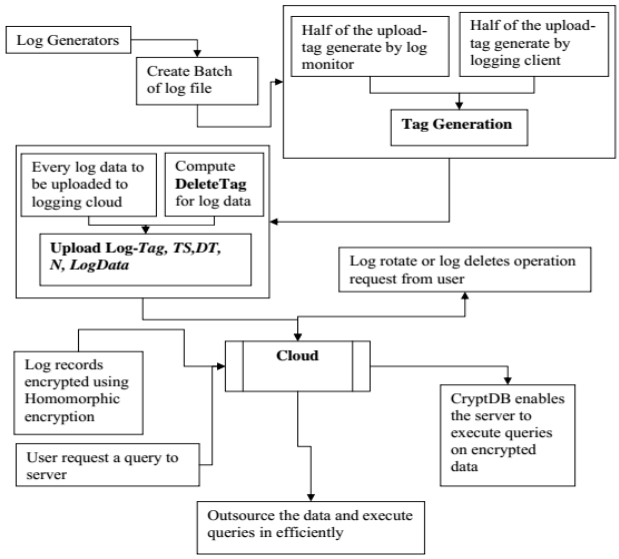


Fig 3.1 - A Tor non-exit relay with a maximum output of 239.69 KB/s

IV. PROPOSED SYSTEM

The main function of CryptDB's proxy is to store the secret master key MK, the database schema, and the current encryption layers of all columns. The DBMS server sees an anonymized schema (where the table and column names are being replaced by the opaque identifiers), encrypted user data, and some auxiliary tables used by CryptDB. CryptDB helps the server with for specific user-defined functions (UDFs) that enables the server to compute it on cipher texts for certain operations.



4.1 The Overall Architecture of a System

Processing a query in CryptDB involves four steps:

1. The application issues a query, which the proxy intercepts and rewrites: it anonymizes each table and column name, using the master key MK, encrypts each constant in the query with an encryption scheme best suited for the desired operation.

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

2. The proxy checks if the DBMS server should be again given the keys to adjust the encryption layers before executing the query, and if it is so, issues an UPDATE query at the DBMS server which invokes a UDF for adjusting the encryption layer to the appropriate columns.
3. The proxy helps in forwarding the encrypted query to the DBMS server, where the query is being executed using the standard queries.
4. The DBMS server helps in returning the (encrypted) query result, where the proxy decrypts and returns the decrypted content to the application. Homomorphic encryption (HOM) is a secure probabilistic encryption scheme, which allows server to perform the required operation on the encrypted data with the final result being decrypted at the proxy.

Advantages:

- Dynamically adjusting the encryption level is done by using much of encryption for minimizing the information that are revealed to the untrusted DBMS server.
- Highly secure encryption schemes.
- Reduce the communication overhead between log monitor and the logging cloud .

V.RESULTS AND DISCUSSIONS

Fig.5.1 shows the amount of information that would be revealed to the adversary in practice, we examine the steady-state onion levels of different columns for a range of applications and queries. To quantify the level of security, we define the MinEnc of a column to be the weakest onion encryption scheme exposed on any of the onions of a column when onions reach a steady state (i.e., after the application generates all query types, or after running the whole trace).

In the graph representation:

- X-axis represents the files range from 0 to 9 and
- Y-axis represents security level of the system.

From the results the proposed system is much efficient than the existing implementations.

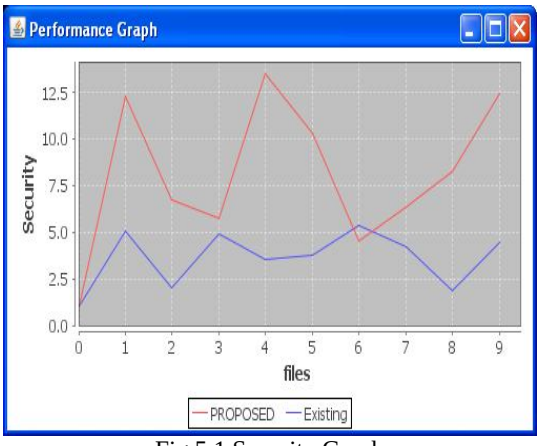


Fig.5.1 Security Graph

Fig.5.2 shows the comparison between the computational time of existing and proposed algorithm.
In the graph

- X-axis represents the files range from 0 to 9 and
- Y-axis represents the computation time for run a algorithm in varies from 0 to 100 seconds.

From the results obtained the proposed algorithm runs faster than the existing algorithm.

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

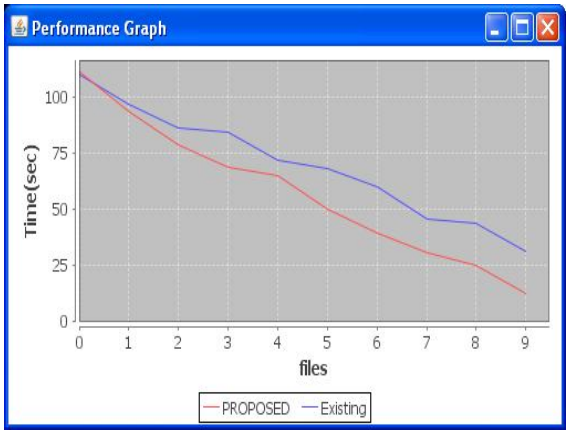


Fig 5.2. Time Graph.

V. CONCLUSIONS

The proposed system includes the homomorphic encryption scheme for encrypting log records. In this case, the cloud that has been logged out can execute some queries which is on the encrypted logs without any further excludence in confidentiality or privacy. A system which provides a practical and strong level of confidentiality in facing two significant threats and which confronts database-backed applications are as follows:

1. Curious DBAs and
2. Arbitrary compromises of the application server and the DBMS.

CryptDB aims three ideas & meets them:

1. Running queries efficiently over encrypted data by using a novel encryption strategy, dynamically helps in adjusting the encryption level by using more number of encryption schemes to minimize the information that are being revealed to the untrusted DBMS server, and by chaining the encryption keys to user passwords in such a way that it allows only authorized users to gain access to the encrypted data.

The current implementation of the client in the logging system is based on loose coupling with the operating system. The future enhancement, includes refining the log client implementation by tightly integrating with the OS such that the current logging process is replaced.

REFERENCES

[1] Dieudonne Mulamba, Indrajit Ray, Mariappan Rajaram, Mikhail *Strizho "Secure Logging As a Service—Delegating Log Management to the Cloud"* IEEE systems journal, vol. 7, no. 2, june 2013.

[2] K. Kent and M. Souppaya. " *Guide to Computer Security Log Management", NIST Special Publication 800-92.* sep.2006

[3] PCI Security Standards Council.*Payment Card Industry (PCI) Data Security Standard*— Requirements and Security Assessment Procedures Version 2.0 Oct 2010.

[4] Sarbanes-Oxley Act 2002." *A Guide to the Sarbanes-Oxley Act"* Sep. 2002.

[5] C. Lonvick, "*The BSD Syslog Protocol"*, Request for Comment RFC 3164,Internet Engineering Task Force, Network Working Group, Aug. 2001.

[6] D. New and M. Rose, "*Reliable Delivery for Syslog"*, Request for Comment RFC 3195, Internet Engineering Task Force, Network Working Group, Nov. 2001

[7] M. Bellare and B. S. Yee, "Forward integrity for secure audit logs," Dept.Comput. Sci., Univ. California, San Diego, Tech. Rep., Nov. 1997.

[8] Bellare, R. Canetti, and H. Krawczyk." Keying hash functions for message authentication". In N. Koblitz, editor, Advances in Cryptology: Crypto '96 Proceedings, volume 1109 of Lecture.

[9] D. Ma and G. Tsudik, "A new approach to secure logging," *ACM Trans. Storage*, vol. 5, no. 1, pp. 2:1–2:21, Mar. 2009.

[10] Swanson, M., Guttman, B.:" Generally accepted principles and practices for securing information technology systems". In: NIST 800-14. 1996

[11] Bellare, M., Yee, B.: "Forward integrity for secure audit logs". In: Technical Report, Computer Science and Engineering Department, University of San Diego. Nov. 1997

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

[12] Bellare, M., Yee, B.: "Forward-security in private-key cryptography". In: Proc. of CT-RSA'03.
[13] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inform. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.
[14] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. Nat. Comput. Conf.*, p. 313 Jun. 1979.
[15] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: How to cope with perpetual leakage," in *Proc. 15th Ann. Int. Cryptology Conf.*, pp. 339–352. Aug. 1995
[16] K. Nørv°ag, O. Sandst°a, and K. Bratbergsengen, "Concurrency control in distributed object oriented database systems," . *Symp. Adv. Databases Inform. Syst.*, , pp. 32–32Sep. 1997.
[17] B. Schneier and J. Kelsey, "Security audit logs to support computer forensics," *ACM Trans. Inform. Syst. Security*, vol. 2, no. 2, pp. 159–176.
[18] R. Anderson, \Robustness Principles for Public Key Protocols," Advances in Cryptology \CRYPTO '95, Springer-Verlag,pp. 236 ,247 1995.
[19] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11,pp. 612–613, Nov. 1979.
[20] G. R. Blakley, "Safeguarding cryptographic keys," in *Proc. Nat. Comput. Conf.*, Jun. 1979, p. 313.
[21] R. Ostrovsky and M. Yung, "How to withstand mobile virus attack,"in *Proc. 10th Ann. ACM Symp. Principles Distributed Comput.*, Aug.1991,pp. 51–59.
[22] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung, "Proactive secret sharing or: How to cope with perpetual leakage," in *Proc. 15th Ann. Int.Cryptology Conf.*, Aug. 1995, pp. 339–352.
[23] I. Teranishi, J. Furukawa, and K. Sako, "*k*-times anonymous authentication(extended abstract)," in *Proc. 10th Int. Conf. Theor. Appl.Cryptology Inform. Security*, LNCS 3329. 2004, pp. 308–322.
[24] D. L. Wells, J. A. Blakeley, and C. W. Thompson, "Architecture of an open object-oriented database management system," *IEEE Comput.*, vol. 25, no. 10, pp. 74–82, Oct. 1992.
[25] K. Nørv°ag, O. Sandst°a, and K. Bratbergsengen, "Concurrency control in distributed object oriented database systems," in *Proc. 1st East-Eur. Symp. Adv. Databases Inform. Syst.*, Sep. 1997, pp. 32–32.
[26] R. Droms, *Dynamic Host Configuration Protocol*, Request for Comment RFC 2131, Internet Engineering Task Force, Network Working Group, Mar. 1991.
[27] Project: AN.ON—Anonymity Online. (2012, Mar.). *JAP Anonymity and Privacy* [Online]. Available: http://anon.inf.tu-resden.de/index−en.html
[28] Ultrareach Internet Corporation. (2012, Mar.). *Ultrasurf—Privacy, Security,Freedom* [Online]. Available: http://ultrasurf.us/index.html
[29] Global Internet Freedom Consortium. (2012, Mar.). *FreeGate* [Online].Available: http://www.internetfreedom.org/FreeGate