2011 3rd International Conference on Cyber Conflict
C. Czosseck, E. Tyugu, T. Wingfield (Eds.)
Tallinn, Estonia, 2011 © CCD COE Publications

Preserving Organizational Privacy in Intrusion Detection Log Sharing

Hayretdin Bahşi
Turkish National Research Institute of Electronics and Cryptology
Kocaeli, Turkey
e-mail: bahsi@uekae.tubitak.gov.tr

Albert Levi
Faculty of Engineering and Natural Sciences
Sabancı University
İstanbul, Turkey
e-mail:levi@sabanciuniv.edu

Abstract- **This paper presents a privacy-preserving framework for organizations that need to share their logs of intrusion detection systems with a centralized intrusion log management center. This centralized center may be an outsourced company that provides an intrusion detection management service to organizations or a system of the National Computer Emergency Response Team that probes the attacks targeting organizations that have critical information systems. For reasons of ensuring privacy, we adopt the notion of *l*-Diversity in the course of collecting intrusion logs from organizations. Within our framework, an organization ensures the people in the center cannot discern the exact origin of any intrusion log among the other *l*-1 organizations. Also, it is not possible to precisely identify the classification type of an intrusion log from among other *l*-1 types. Within this framework, the intrusion log management center can analyze the anonymous data, since the proposed privacy preserving solution creates little information loss. If required, it sends an alarm to the appropriate organization within a reasonable time. The center has the option of publishing useful information security statistics about specific organizations or about the whole ecosystem by using the privacy preserved intrusion logs.**

Keywords: privacy preserving framework, intrusion detection, log sharing

I. INTRODUCTION

It is known that hackers share information with each other in order to attack victims. In underground communities, zero day vulnerability information, target victim information, stolen credit card numbers, bots, spam mail lists, attack tools, etc. are shared or sold easily. On the other hand, system managers, who strive to defend their systems against hackers, need to share related materials about defensive tools, methods and information. The defensive experience of an organization can easily be transferred to others by sharing intrusion detection system logs.

It is common that most of the organizations somehow use intrusion detection systems to detect attacks against their systems. These systems do not always produce useful outputs. In particular, the elimination of false positive alarms

requires labor intensive work. An information security expert has to choose the set of attack signatures that are appropriate for his system and eliminate false positive alarms. However, most organizations cannot reserve staff for this task due to a lack of specialized technical personnel or due to a lack of budgetary funding. Under these circumstances, the outsourcing of intrusion log analysis could be a good alternative.

Nowadays, National Computer Emergency Response Teams (NCERT) are determining ways to perform proactive nationwide security countermeasures in order to detect and prevent cyber attacks targeted at national critical information infrastructures. These infrastructures generally belong to different organizations. NCERTs try to determine ways to centrally probe them. Probing aims to deduce the overall threat state of each organization and determine the overall threat level of the country. For this aim, a distributed intrusion detection system has to be setup and managed. Moreover, collected intrusion logs have to be centrally stored and analyzed.

In the above both cases, there is a need for a central intrusion log management office (CILMO) to store logs of different organizations centrally, analyze them, detect attacks, send alarms to organizations and generates statistics for determining nationwide threat levels.

The primary obstacle in forming a CILMO is the privacy concerns of organizations. Intrusion logs contain valuable information about organizations, such as detailed knowledge of targeted information assets, attack times, types of attacks, results of attacks, etc. Organizations are reluctant to share intrusion logs due to two main reasons. First, they do not fully trust the personnel of CILMO, because administrators of CILMO may intentionally misuse their attack information. The second reason may be the lack of appropriate security and privacy countermeasures, which have to be applied to the intrusion logs during their transmission, processing and storage. Without solving these security and privacy problems, organizations generally do not wish to send their intrusion logs to a CILMO, even though it may have been set up by a NCERT team.

Organizations are confronted with the dilemma between privacy risks and the benefits of sharing intrusion logs. Therefore, one has to deal with the trade-off between privacy and information loss, according to the needs of organizations.

In this paper, a privacy-preserving framework based on *l*-diversity is presented for intrusion log sharing. This notion guarantees that the exact classification type of an intrusion log cannot be identified among other *l*-1 types. Also, privacy schema enables us to hide the source organization of the intrusion log among *l*-1 organizations. Through the collection of privacy-preserved intrusion logs, this framework enables CILMOs to perform detailed security analysis of organizations, draw conclusions about the general security status of organization categories and prepare a warning mechanism.

The general structure of the paper is as follows: Section II gives some background information and introduces the threat and network model. Section III details the proposed anonymization method. Section IV gives the results of experiments performed in evaluation of the proposed method. Section V concludes the paper.

II. MOTIVATION AND BACKGROUND

A. *k-Anonymity and l-diversity*

Privacy problem cannot be easily solved by merely removing identity information (name, social security number, etc.) from the records of individuals. Data fields called quasi-identifiers may be used to identify a person by using external information sources. This attack technique is called “Re-identification attack” [1] or “record linkage” [2]. For example, in a hospital database, address, sex or other attributes can precisely identify an individual. *k*-Anonymity [1], which is defined as being not identifiable of an individual within a set of *k*-1 individuals, is used as a privacy criterion in order to make data resistant to re-identification attacks. *k*-Anonymity generalizes or suppresses quasi-identifiers of data records so that an individual cannot be differentiated between other records of *k*-1 individuals by using those quasi-identifiers.

It has been shown that without finding the exact owner of a record, if sensitive attribute exists in a record, it may be possible to identify the sensitive attribute of an individual in some circumstances by an attack called an “attribute linkage attack" [2]. Sensitive attribute includes information such as the health of a patient in a hospital database. In order to prevent this problem, *k*-anonymity notion extended in some studies. Machanavajjhala et. al. extended *k*-anonymity with a *l*-diversity notion in order to cover these attacks [3]. In addition to *l*-diversity notion, *p*-sensitivity and *t*-closeness notions are proposed [4], [5].

B. *Threat and Network Model*

In our study, organizations send their intrusion logs to a trusted party. In a realistic scenario, a trusted party may be an Internet service provider (ISP). Normally, all the network traffic between the Internet and organizations is managed by ISPs. Organizations legally protect themselves against the possible malicious activities of ISP administrators by service level agreements, which include non-disclosure and security protection terms. ISPs can be presumed to be trusted parties due to these agreements.

A sample system topology for the proposed privacy framework is given in Figure 1. A trusted party anonymizes intrusion logs, strips off the destination IP information of a log and appends a destination tag instead of the destination IP, which only represents the source organization. Target Service, source IP and detection time attributes are classified as quasi-identifiers and intrusion classification is accepted as sensitive attribute. According to this attribute classification, our anonymization method provides the prevention of record and attribute disclosure by providing *l*-diversity property of intrusion logs.

It is assumed that in each log originating from organizations, pre-exploitation, exploitation and post-exploitation activities are correlated and one log entry is created for each attack. If one attack targets the many servers of an organization, only one log entry is produced by IDS.

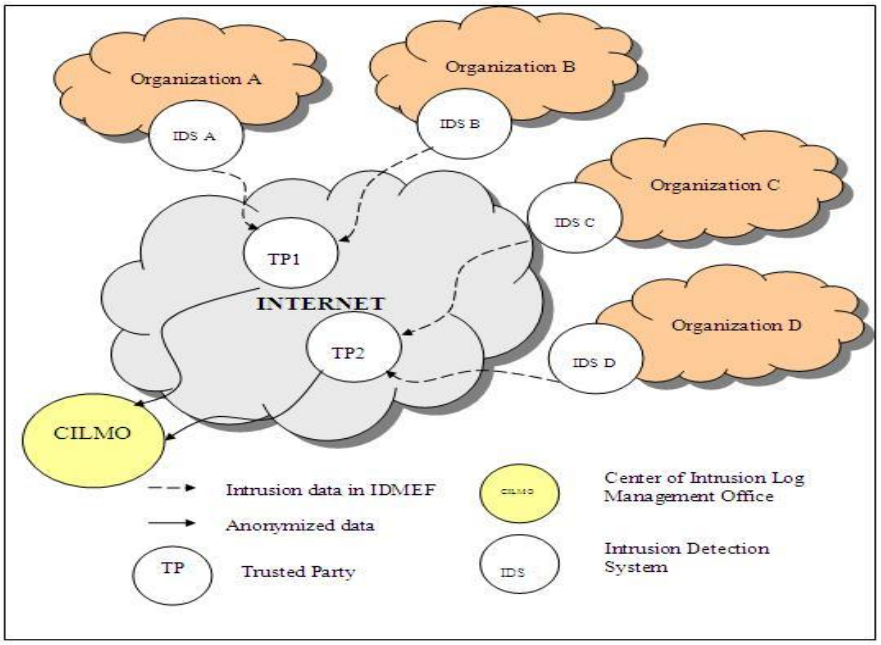

Figure 1. Figure 1. System Topology for Privacy Framework

C. *Related Work*

Some organizations implement intrusion log collection systems for determining the general security level of the Internet. Deepsight Threat Management System [6], which is managed by Symantec, gives information to its customers about emerging threats, vulnerabilities, risks, etc. The system does not use any anonymization method during data collection. Internet storm center, which is implemented by SANS [7], collects intrusion detection system and firewall logs from volunteer organizations producing general analysis results for the public, and creates customized warning information for organizations. They just simply remove the identifying parts of intrusion data by masking the destination IP of logs.

There are studies about anonymzing the IP address of network logs. In a basic solution, actual IP addresses are replaced by a randomly selected IP addresses according to a permutation function. New random IP addresses do not even contain the sub-net information.

Truncation is another anonymization method that converts a fixed number of the least significant bits of an IP address to zero. This means that the remaining information can show only the subnet or network class information of IP addresses. From anonymized data, anyone can deduce the subnet information but cannot determine whether logs belong to a particular subnet.

In *prefix-preserving pseudonymization*, which is adapted in *TCPdriv* [8], IP addresses are mapped to pseudorandom anonymized IP addresses by an

anonymization function that uses common tables. Reference [9] proposed a prefix-preserving pseudonymization method, Crypto-PAn, which works consistently in multiple traces by using a shared key. Crypto-PAn is re-implemented in Java for the anonymization of Netflow logs [10]. The anonymization of all fields of Netflow and syslog data for sharing them with managed security service providers is performed in [11].

III. PROPOSED ANONYMIZATION METHOD L-ACM

k-ACM (*k*-Anonymous Clustering Method) is proposed in [12], which k-anonymizes the data by using the hierarchical bottom-up clustering method. This method is applied for the anonymization of collected data in wireless sensor networks [12], [13]. In this paper, *k*-ACM is modified for the anonymization of intrusion logs in order to make them *l*-diversity. The proposed method is referred to as the *l*-diversity Anonymous Clustering Method (*l*-ACM).

Subsection III.A explains how the collected information is represented in our proposed method. In Subsection III.B, distance metric, which is used in the clustering process, is described. Subsection III.C presents details on the bottom-up clustering process, which is the core of the proposed method.

A. Data Representation

l-ACM uses the data representation model used in [12], [13]. This subsection describes the details of this model. Suppose input data is a table T with m attributes, r records. Tij represents the j'th attribute of the i'th record where $\{i : 1 \leq i \leq r\}$ and $\{j : 1 \leq j \leq m\}$. Table T is represented by a set of bit strings B, where Bij is a bit string representation of j'th attribute of i'th record. The k'th bit of Bij is shown as $Bij(k)$.

Suppose that the j'th attribute of a table is categorical and there are dj distinct values. These values are indexed by k and shown as $Vj(k)$ where $\{ k : 1 \leq k \leq dj \}$. The bit string of this categorical attribute has a size of dj and is formed as follows:

$$\text{If } Tij = Vj(k) \text{ then } Bij(k) = 1 \text{ else } Bij(k) = 0 \text{ as } \forall k : 0 \leq k \leq dj ,$$

If the attribute is numerical, the range of the attribute is divided into equal-sized intervals and each interval constitutes a categoric value.

B. Information Loss Metric

In order to evaluate the quality of data, *l*-ACM uses the information loss metric of *k*-ACM [12]. This metric basically uses the entropy concept of the information theory [14]. Information loss is quantified by the difference of entropies between the *l*-diversified data and the original data.

Assume that input data, T, has r records and m attributes. B is the bit string representation of data set, T. C is the random variable that gets the probability value of an attribute value in a *l*-diversified data entry, which is the actual attribute value in the original data. B is normalized by the number of bits that have

value “1” (from here on we use “true bit” to refer to a bit that has the value “1”). Normalized version forms data set $\overline{B}$.
Information loss of a data table *T*, *IL*(*T*), is equal to the conditional entropy, $H(C \mid B)$. Here, the conditional entropy gives the uncertainty about the prediction of the original attribute values of a record when we have the knowledge of corresponding *l*-diversified bit strings of that record. The original data has only one true bit. Anonymization increases the number of true bits. Each true bit actually represents the possible original attribute value. As the number of true bits increases, disorder of the data increases because it is harder to predict which one of them is the original true bit. Conditional entropy $H(C \mid B)$, which is equal to the information loss of table *T*, *IL*(*T*), can be determined as follows:

$$\begin{aligned} IL(T) = H(C \mid B) &= \sum_{B_{ij} \in B} p(B_{ij}) H(C \mid B = B_{ij}) \\ &= - \sum_{B_{ij} \in B} p(B_{ij}) \sum_{k \in \{1..z\}} p(C = k \mid B_{ij}) \log p(C = k \mid B_{ij}) \end{aligned} \quad (1)$$

In Eqs. (1), it is assumed that each attribute is converted to bit strings of the size *z*. This means that all categorical attributes have *z* distinct attribute values and all numerical attributes have *z* number of interval ranges. Also, it is assumed that all *k*’s, where the equalities of $p(C=k|Bij)=0$ are true, are excluded from the summation. *C* random variable can take values from the set $\{1..z\}$. Actually, $\overline{B}$ is calculated for determining the value of this random variable.

$$p(C = k \mid B = B_{ij}) = \overline{B}_{ij}^{k} \; for \; each \; k : 1 \le k \le z \quad (2)$$

In Eqs. (2), it is assumed that each record has equal probability to be chosen and each attribute of record has the same probability. Therefore, the probability mass function of the *j*’th attribute of the *i*’th record, $p(Bij)$ is calculated as $p(Bij)=1/m.r$. Eqs (1) can be rewritten as follows:

$$IL(T) = - \sum_{B_{ij} \in B} \frac{1}{m.r} \sum_{k \in \{1..z\}} \overline{B}_{ij}^{k} . \log(\overline{B}_{ij}^{k}) \quad (3)$$

Suppose that *F* is the array that contains the number of true bits of the bit string array *B*. The total number of true bits in Bij is Fij. The total number of elements in $\overline{B}ij(k)$ that have the value of $1/F_{ji}$ is equal to F_{ji}, and the rest are zero. Therefore, the second sum operation of Eqs. (3) yields the value, $logl/F_{ji}$. The

simplest equation for the information loss of data table T, $IL(T)$, can be calculated as follows:

$$IL(T) = -\sum_{F_{ij} \in F} \frac{1}{m.r} \log\left(1/F_{ij}\right) == \frac{1}{m.r} \sum_{F_{ij} \in F} \log(F_{ij}) \quad (4)$$

C. *Bottom-up Hierarchical Clustering Process*

Method bases on forming clusters of input vectors iteratively. Each cluster numerated as C_j^l in each epoch, l, contains a number of input vectors, N_j^l, and a representative vector, R_j^l where j is the index number of cluster. Suppose that the k^{th} data item of the representative vector is denoted as $R_j^l[k]$. The representative vector is actually the anonymized output of input vectors belonging to the cluster that is formed by generalization operations of some data parts of vectors.

The hierarchical clustering process begins with the assumption that each input vector constitutes a separate cluster and that vector is also a representative vector of the cluster. In each epoch, by using the information loss metric described in Section III.B, distances between each cluster are calculated. The distance between any two clusters is actually equal to the information loss that may occur if both clusters are merged.

The two clusters that have the smallest distance, e.g. clusters C_s^l and C_t^l, are chosen for merging. The new bigger cluster, C_u^{l+1} which contains the vector items of both clusters, is formed and the former two clusters are deleted. N_u^{l+1} is equal to the sum of N_s^l and N_t^l . Anonymization is performed by generalization. $R_u^{l+1}[k]$ is equal to the XOR of $R_s^l[k]$ and $R_t^l[k]$.

l-ACM keeps on clustering iterations up to the point where each cluster contains a record set that has distinct l sensitive attribute values and l different sets of quasi-identifier attributes. Representative vectors of remaining clusters form the l diversified outputs.

A target organization can be considered as an identifier of an intrusion log. In our case, CILMO needs the names of the target organizations in order to perform the required security analysis tasks. The names of the target organizations are transferred to CILMO in such a way so that nobody can deduce the name of the exact organization of an intrusion log among the l-1 organizations.

The same organization may send many intrusion logs to CILMO. If one anonymity set produced by l-ACM has many intrusion logs of the same organization, this situation may violate l-diversity property. Therefore, l-ACM guarantees that each

record in each cluster has to belong to a different organization. Suppose that n is the number of records. The set of all target organizations is represented as $\{O_1, O_2, ..., O_n\}$. Assume that all records have m different sensitive attribute values where $m > l$ and these attributes values are $\{ S_1, S_2, ..., S_m \}$. The data sent to CILMO can be shown as $\{O_1, O_2, ..., O_n\}, R_i, \{ S_1, S_2, ..., S_m \}$.

A running example of l-ACM is shown in TABLE I and TABLE II. Assume that each destination IP belongs to a different organization. The destination IP of the intrusion log is replaced with the name of the organization during anonymization. The trusted party gathers the original data shown in TABLE I, produces three clusters that each have two elements and makes the data 2-diversified. Each row in this table represents one cluster. All the attributes are converted to sets of distinct attribute values. l-ACM guarantees that in the destination organization attribute, two distinct organization names exist and the classification attribute consists of a set that has two different classification values. Since the source IP, time and destination port attributes are chosen quasi-identifiers, l-ACM tries to minimize the number of distinct attribute values of these attributes in anonymized output.

TABLE I. AN EXAMPLE OF THE ANONYMIZATION OF INTRUSION LOGS – ORIGINAL DATA

Dst IP	Src IP	Time	Dst Srv	Classification
201.2.1.10	195.100.4.4	11:00	53	DNS Zone Transfer
223.23.5.4	195.100.4.4	11:30	8080	WEB IIS ISAPI
212.125.12.12	198.166.3.3	11:40	3372	DoS MSDTC
222.19.1.103	190.67.30.3	11:45	1543	NETBIOS SMB
208.234.3.105	199.201.45.56	11:55	80	WEB-COLDFUSION
200.188.5.17	191.34.32.1	12:05	1548	DOS IGMP

TABLE II. AN EXAMPLE OF THE ANONYMIZATION OF INTRUSION LOGS – 2-DIVERSIFIED DATA

Dst IP	Src IP	Time	Dst Srv	Classification
{ O1, O2}	{195.100.4.4}	{11:00, 11:30}	{8080, 53}	{DNS Zone Transfer, WEB IIS ISAPI}
{ O3, O4}	{198.166.3.3, 190.67.30.3}	{11:40, 11:45}	{1543, 3372}	{DoS MSDTC, NETBIOS SMB}
{ O5, O6}	{199.201.45.56, 191.34.32.1}	{11:55, 12:05}	{80, 1548}	{WEB-COLDFUSION, DOS IGMP}

D. *Warning Mechanism*

CILMO may need to warn organizations about a very critical intrusion. Likewise, if the proposed anonymization method is used in intrusion log sharing, CILMO does not know the exact intrusion classification for the exact originator. It only knows that a set of organization corresponds to a set of intrusion classification values. CILMO may be interested in one intrusion classification among these values. If it is assumed that the trusted party does not store any information including the mappings of original data with anonymous data, the warning can be performed by only distributing it to each IDS management server of all candidate

organizations. The details of the warning mechanism are described through an example in Figure 2. Each organization sends their logs, which are labelled as r_1, $r_2, .., r_6$ to the trusted party (TP), in step 1. TP anonymizes the data according to 2-diversity criteria and sends the anonymous outputs a_1, a_2, a_3 to CILMO in step 2. Assume that CILMO decided to warn the organizations about the DNS Zone Transfer attack due to its seriousness. Assume that r_1 has this classification type. CILMO chooses the anonymous record (a_1) which has this attack type from the set of classification attributes. CILMO creates w_1 from a_1 by stripping off all organization attributes and all classification information except "DNS Zone Transfer" and sends w_1 to IDS management servers of organization 1 (O1) and organization 2 (O2) in steps 3 and 4. In step 5, O1 and O2 query whether an intrusion log exists about the profile given in w_1 and determine whether the corresponding warning is related with their organization.

A drawback of this mechanism is that the organization O2, which decides the warning, does not belong to itself in the above example. It also receives the profile information of the intrusion that occurred for O1 without knowing the targeted organization is O1.

If the trusted party is allowed to store mapping information between original data and anonymous output, after deciding the warning message, CILMO sends w_1 to TP. TP finds the exact intrusion log record that matches with w_1, deduces that it is r_1 and relays r_1 to O1. In this method, an organization does not learn anything about the intrusion logs of other organizations. The warning is sent directly to the owner organization of the intrusion log.

IV. PERFORMANCE EVALUATION OF L-ACM

In this part, the performance of *l*-ACM is evaluated in terms of *information loss* and the *average response time* of intrusion log records. The average response time, T_{avg}, shows the average amount of times between the generation of the log at the owner organization and the arrival of the corresponding warning to that organization from CILMO.

In our experiments, each organization generates an intrusion log in a such a way that all the attributes of logs are formed using uniform distribution. The log generation time for i^{th} log record is represented as t_g^i. Log generation rate, *lgr*, which is the number of produced logs per minute, is a predetermined parameter that adjusts the speed of log generation. It is assumed that each organization uses the same log generation rate. All log records generated in one minute are collected at the organization site and they are sent to CILMO at the end of that minute. Therefore, i^{th} log record waits 60 - t_g^i seconds at the organization site before being sent to CILMO. After CILMO receives the logs, the anonymization operations take place by using *l*-ACM. Anonymization is completed in several steps. In each step, the data set that includes only one record from each organization is chosen among the received logs and they are anonymized.

Otherwise, if we include more than one record from each organization, an anonymity set may contain more than one record belonging to same organization, which violates *l*-diversity property. The restriction of one record from the same organization actually means that the number of steps needed for completion of anonymization is numerically equal to the log generation rate. The duration of the m^{th} anonymization step is represented as t_a^m.

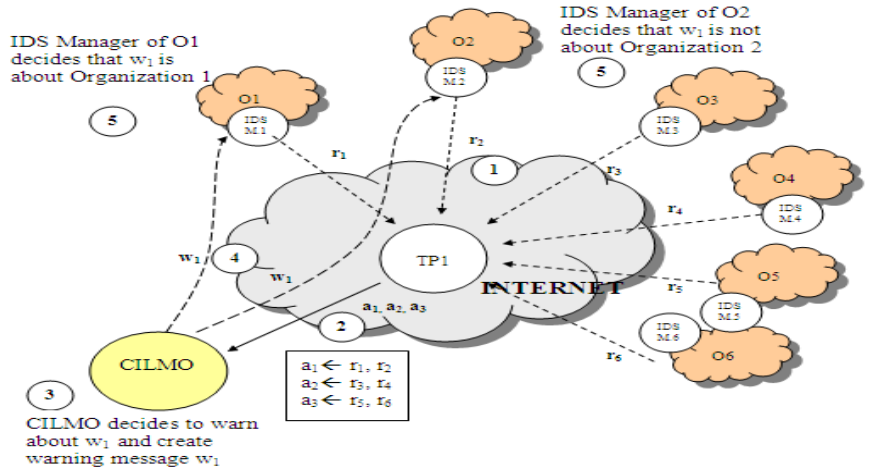

Figure 2. Figure 2. Warning Mechanism with the requirement that trusted party does not store any information

In *l*-ACM, we use a record selection method for preparing the input data of each anonymization step. Our method chooses an initial record from the first organization. For each other organization, the logs of an organization are compared with the record of the first organization and the one that bears the most similarity is chosen as an input record in that step.

Anonymized outputs are analyzed by CILMO. If analysis results require the sending of a warning to the appropriate organization, warnings are sent by using one of the methods given in Section III.D. In performance calculations, a parameter called log analysis time, t_l, is used for the log analysis of one log record at CILMO. Warnings are sent after this analysis time has passed.

The transmission time needed for transferring one log record from the organization to CILMO and the time for transferring one warning to the organization is represented as t_r. In average response time calculations, we assume that for each log record, CILMO sends a warning message. The average response time for a log record is calculated as given in Eqs. (5). We assume that the total number of the input record is *n*.

$$T_{avg} = (60 - t_g^i) + \sum_{s=1}^{s=m} t_a^m + t_l + 2.t_r \quad (5)$$

The effects of changes in parameter *l* and *lgr* with respect to information loss and response time performances of *l*-ACM, are investigated via simulations. Experiments are performed in a laptop that has 1.20 GHz CPU and 2GB RAM. Intrusion data is synthetically generated. A java implementation is developed for data generation, the application of *l*-ACM and evaluating the results.

k-ACM calculates the information loss according to Eqs. (4). In this formula, F_{ij} is the total number of bits that have the value of '1' for the *i*th record of *j*th attribute. On the other side, *l*-ACM produces anonymized output with an attribute value sets instead of bit strings. Therefore, *l*-ACM uses the size of the attribute value set (which means the number of distinct elements in the set) instead of F_{ij} .

There are 100 distinct attackers in the network. The number of distinct values for intrusion classification is 15 and the number of slots for time value is 100. There are 10 distinct destination services in the data set. According to these parameters, maximum information loss is calculated as 5.54 via the help of Equation 4.

The effects of *lgr* and *l* values on information loss results is given in Figure 3. In these experiments, the number of organizations that send their logs to CILMO is fixed to 500. As shown in Figure 3, increase in *lgr* does not affect information loss values for each *l* value. The effects of *lgr* and *l* values on average response time are given in Figure 4. In this experiment, number of organizations is also fixed to 500. It is observed that the average response time increases as *lgr* increases for each *l* values. There is a linear relationship between the average response time and *lgr* values. Since *lgr* also determines the number of anonymization steps performed at CILMO, an increase in the number of steps increases the time for anonymization operations. For the same *lgr*, we get higher than average response time values for higher *l* values due to the need for much more processing in hierarchical clusterings.

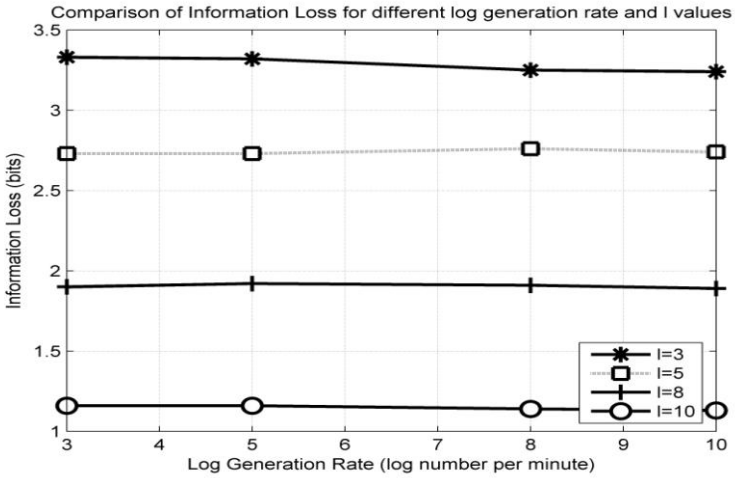

Figure 3. Figure 3. Effects of *lgr* and *l* on Information Loss

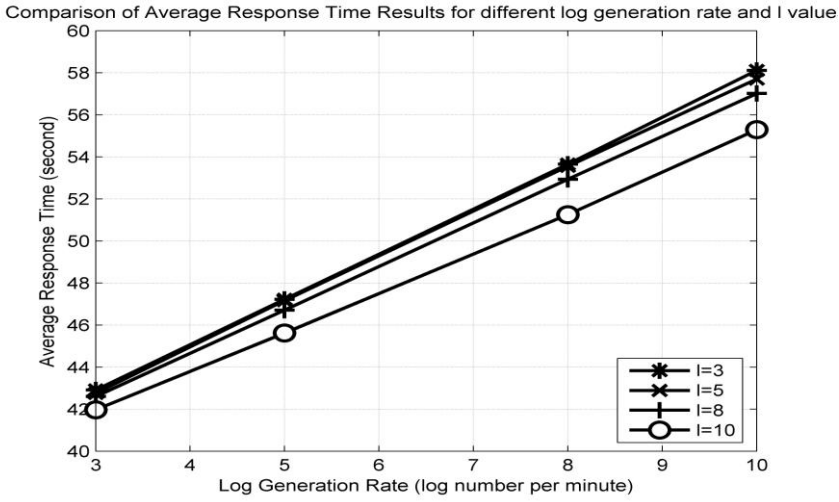

Figure 4. Figure 4. Effects of *lgr* and *l* on Average Response Time

Effects of the changes in the number of organizations are analyzed. Figure 5 shows the effects of organization number to information loss. Figure 6 analyzes the average response time results of *l*-ACM with a different number of organizations. In these experiments, the *l* and *lgr* values are fixed to 5 and 8 respectively.

From Figure 5, it is deduced that the information loss value decreases as the number of organizations increases. Since, anonymization is performed among bigger sets of log records in higher organization numbers; *l*-ACM has the possibility to find more similar records during hierarchical clustering. However, the decrease is very small according to experimental results.

Figure 6 shows that a higher number of organizations cause higher response times. There exists an exponential increase in response times. An increase in the number of organizations means higher number records are given as an input to *l*-ACM in each anonymization step.

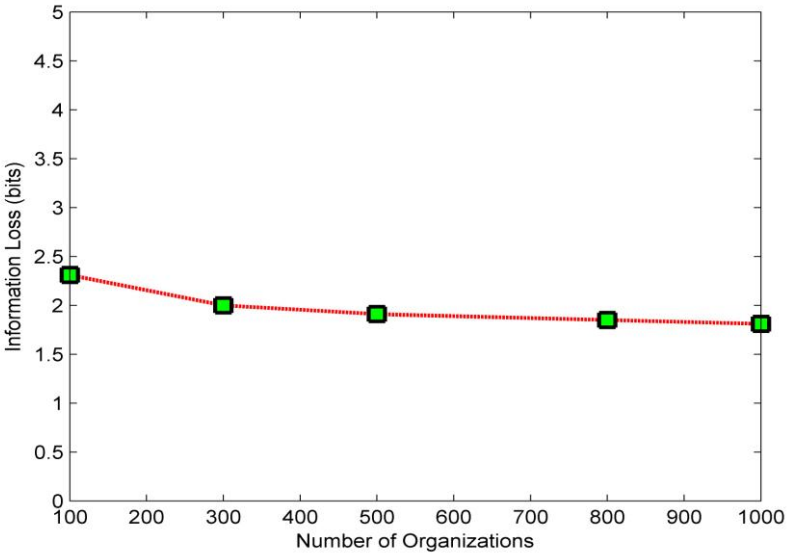

Figure 5. Figure 5. Effects of Organization Number on Information Loss

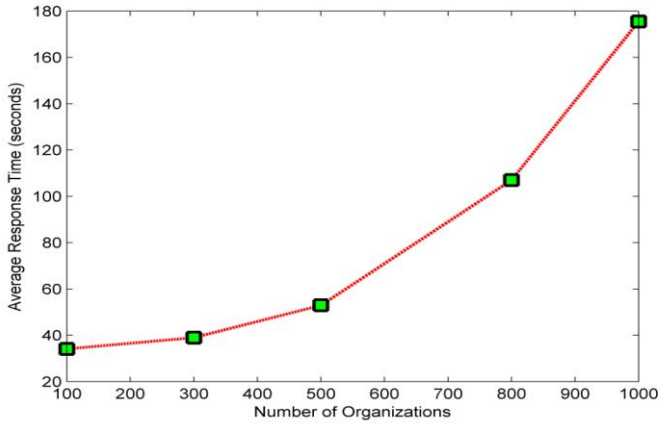

Figure 6. Figure 6. Effects of Organization Number on Average Response Time

V. CONCLUSION

In this paper, the privacy preserving framework is proposed for the collection of intrusion logs from different organizations through a central intrusion log management office. This office is tasked for determining the overall security posture of the whole organization ecosystem, the designation of the security status of monitored organizations, and it gives feedback or warnings to organizations about critical intrusions. The privacy threat model states that the collected log has to have *l*-diversity property. This means, any administrator of the central office cannot deduce the exact classification type of intrusion log among the *l* classification types. *l*-ACM (*l*-Diversity Anonymous Clustering Method), is proposed for this purpose. Different warning mechanisms are presented according to the security requirement on whether trusted parties are allowed to temporarily store network traffic.

REFERENCES

[1] L. Sweeney, "k-anonymity: A model for protecting privacy," Int'l Journal on Uncertainty, Fuziness, and Knowledge-based Systems 10(5),

[2] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey on recent developments," *ACM Computing Surveys* , 2009.

[3] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "l-diversity: Privacy beyond k-anonymity," in Proceedings of 22nd International. Conference on Data Engineering, p. 24, ICDE, 2006.

[4] M. T. Truta and V. Bindu, "Privacy protection: *p*-sensitive *k*-anonymity property," in *Proceedings of the Workshop on Privacy Data Management*, p. 94, Workshop on Privacy Data Management, In Conjunction with 22th IEEE International Conference of Data Engineering (ICDE), (Atlanta, Georgia), 2006.

[5] N. Li, T. Li, and S. Venkatasubramanian, "*t*-closeness: Privacy beyond *k*-anonymity and *l*-diversity," CERIAS Tech. Report 2007-78, Purdue University, 2007.

[6] "Deepsight threat management system." https://tms.symantec.com/Default.aspx.

[7] "Internet storm center." http://isc.sans.org/.

[8] G. Minshall, "Tcpdriv command manual," 1996.

[9] J. Xu, J. Fan, M. H. Ammar, and S. B. Moon, "Prefix-preserving ip address anonymization: Measurment-based security evaluation and a new crypthography-based scheme," IEEE International Conference on Network Protocols, 2002.

[10] A. Slagell, Y. Li, and K. Luo, "Sharing network logs for computer forensics: A new tool for the anonymization of netflow records," Computer Network Forensics Research Workshop, held in conjuction with IEEE SecureComm, 2005.

[11] J. Zhang, N. Borisov, and W. Yurcik, "Outsourcing security analysis with anonymized logs," 2nd IEEE Intl. Workshop on the Value of Security through Collab., 2006.

[12] H. Bahsi and A. Levi, "k-anonymity based framework for privacy preserving data collection in wireless sensor networks," *Turkish Journal of Electrical Engineering and Computer Science* 18(2), pp. 241–271, 2010.

[13] H. Bahsi and A. Levi, "Data collection framework for energy efficient privacy preservation in wireless sensor networks having many-to-many structures," *Sensors* 10(9), pp. 8375–8397, 2010.

[14] P. Andritsos and V. Tzerpos, "Software clustering based on information loss minimization," in *Proceedings of 10th Working Conference on Reverse Engineering*, p. 334, WCRE 03, 2003.