Journal of Information Processing Vol.22 No.2 118–129 (Apr. 2014)

[DOI: 10.2197/ipsjjip.22.118]

**Regular Paper**

# SIP Flooding Attack Detection Using a Trust Model and Statistical Algorithms

Noppawat Chaisamran[1,a) ] Takeshi Okuda[1,b)] Youki Kadobayashi[1,c)] Suguru Yamaguchi[1,d)]

Received: May 10, 2013, Accepted: October 9, 2013

***Abstract:*** The IP Multimedia Subsystem (IMS) has been constantly evolving to meet the tremendous rise in popularity of mobile services and Internet applications. Since IMS uses Session Initiation Protocol as the main protocol to control a signal, it inherits numerous known security vulnerabilities. One of the most severe issues is the Denial of Service attack. To address this problem, we introduce an anomaly-based detection system using the Tanimoto distance to identify deviations in the traffic. A modified moving average is applied to compute an adaptive threshold. To overcome a drawback of the adaptive threshold method, we present a momentum oscillation indicator to detect a gradually increasing attack. Generally, anomaly-based detection systems trigger many alarms and most of them are false positives that impact the quality of the detection. Therefore, we first present a false positive reduction method by using a trust model. A reliable trust value is calculated through the call activities and the human behavior of each user. The system performance is evaluated by using a comprehensive synthetic dataset containing various malicious traffic patterns. The experimental results show that this system accurately identified attacks and has the flexibility to deal with many types of attack patterns with a low false alarm.

***Keywords:*** IMS, security, flooding attack, statistical analysis, trust

## 1. Introduction

IP Multimedia Subsystem (IMS) is a standard architecture for the Next Generation Network (NGN) designed by the 3rd Generation Partnership Project (3GPP). It is a global, access-independent and standard-based IP connectivity and service control architecture that provides various types of multimedia services to end-users using common Internet-based protocols [1]. The IMS uses a Session Initiation Protocol (SIP) as the control protocol for multimedia communication. Unfortunately, the SIP service can certainly be the target of a variety of attacks and consequently, IMS also inherits these problems. The most severe attack is the Denial of Service (DoS). This attack aims at denying a legitimate user's access to a service or network resource, or at bringing down the servers offering such services. The IMS is much more susceptible to DoS attacks, compared with any previous telecommunication infrastructure. An attacker can easily launch a DoS attack by flooding IMS servers with an enormous number of SIP messages. According to the 3GPP technical specifications [2], [3], IMS security offers features such as authentication and encryption, but it does not provide any mechanism to protect IMS networks from flooding attacks. We believe that as the world's telecom operators gradually deploy IMS to real world networks, SIP flooding attacks will become serious security threats to the telecom operators' businesses.

[1] Nara Institute of Science and Technology, Ikoma, Nara 630–0192, Japan
[a)] noppawat-c@is.naist.jp
[b)] okuda@is.naist.jp
[c)] youki-k@is.aist-nara.ac.jp
[d)] suguru@is.naist.jp

In general, there are two approaches for anomaly detection: signature-based approaches and anomaly-based approaches. With the former approach, incoming traffic is compared to existing patterns. If they are matched, the detection system will raise an alert. This approach can accurately identify known attacks, but cannot detect new anomalies that will lead to false negative alarms. New ways of exploiting computer networks are being invented every day. It seems obvious that there are many ways of circumventing attack signatures. Since several protocols are used in an IMS, there is a possibility that flooding attacks may be generated by any combination of protocols. An intelligent attacker can always develop attacks that remain undetected by signature-based systems. A number of researchers have argued that it is not difficult for an attacker to evade a signature [4]. SIP is also vulnerable to network anomalies that can be easily mounted by utilizing various SIP traffic generators openly available on the Internet. In contrast with the signature-based approach, an anomaly-based approach builds models that represent normal behavior on the network. If there is a significant deviation detected between the observed behavior and the estimated model, an alarm will be raised. The key value of an anomaly-based detection system is that it can automatically infer attacks which are yet unknown, such as the polymorphic packet flooding attacks, and therefore undetectable by signature-based methods. This will alert the network administrator early, and potentially reduce the damage caused by the new attack. Therefore, our detection system developed in this paper adopts the anomaly-based approach.

We propose a flooding attack detection system that generates alerts based on abnormal variation in a selected traffic flow. We use the Tanimoto distance to quantify the correlations among cho-

© 2014 Information Processing Society of Japan

sen attributes. We also propose an adaptive threshold that is used for detecting a significant deviation of traffic. The adaptive threshold can exhibit good performance for high intensity attacks, but it generally suffers from a special attack pattern: an attacker can completely hide an attack by gradually increasing flooding packets. Detection of this attack is particularly important. Therefore, we propose a momentum oscillation indicator to detect such changes in the traffic. A main drawback of detecting changes in the traffic volume is that the detection accuracy may be degraded if the legitimate volume is dynamic or suddenly increasing. This phenomenon happens easily and frequently in telecommunications. We address this problem by integrating a trust model to filter out a legitimate call from suspected traffic. The trust value of each user is computed from the call activities and human behavior of a user, including call duration, call direction, an interactivity ratio, and the diversity of calls. In case of an unknown caller, Dempster-Shafer Theory is applied to combine all trust paths from the caller and then compute an inferred trust value. This detection system is placed in front of a Proxy Call Session Control Function (P-CSCF) in order to monitor incoming traffic. Since P-CSCF is the first core component to be traversed for any request process, deployment of a detection system here will be very helpful in mitigating DoS attacks against IMS networks. Moreover, because it is a stateless approach, our detection system does not require huge memory capacity to process an incoming packets. This can avoid a bottleneck problem when massive traffic comes to the server. Our experimental results demonstrate that this system can achieve a high degree of accuracy in detecting attacks with low false positives. Without an infrastructure modification, this system can be deployed into any IMS network infrastructure to provide defense against DoS attacks.

The rest of the paper is organized as follows. Section 2 provides an overview of the IMS, then reviews related work that deals with flooding attacks against VoIP and IMS networks. Section 3 presents the proposed detection scheme for flooding attacks. The trust calculation and the trust inference methodology is described in Section 4. An evaluation of the method based on simulations with the operator's statistical information and the results are discussed in Section 5. Section 6 discusses the concern regarding our system. Finally, in Section 7 we describe our conclusions and outline our future work.

## 2. Related Work

### 2.1 Overview of IMS

This section provides a brief overview of the IMS. IMS consists of three main components: the Call Session Control Function (CSCF), Home Subscriber Server (HSS), and Application Servers (AS). CSCF is the primary SIP signaling server that acts as the SIP rendezvous point. The CSCF duties are divided into three categories.

- Proxy CSCF: The P-CSCF is the first contact point for users within the IMS. It controls incoming and outgoing messages between the IMS and end users.
- Serving CSCF: The S-CSCF is the focal point of the IMS as it is responsible for handling registration processes, making routing decisions, maintaining session states and storing service profiles.
- Interrogating CSCF: The I-CSCF provides the external interface to other IMS networks and plays an important role in both inter-carrier calls and roaming.

The HSS is the main data storage for all subscriber and service-related data. It provides a database of user credentials and configurations and identifies the home S-CSCF of the subscribers. Finally, the AS hosts and executes services. An example of AS is the Voice Call Continuity Function (VCC server) that guarantees a call persistence when a mobile phone moves between base stations. Many AS can be installed in an IMS network as necessary to support the users. **Figure 1** shows the basic session flow in IMS. The high-level requirements on IMS are summarized in TS 23.228 [5].

### 2.2 SIP Flooding Attacks Detection Methods

The detection algorithms based on a non-parametric cumulative sum (CUSUM) have been applied in Refs. [6], [7]. The CUSUM algorithm belongs to the family of change point detection algorithms that is used for detecting changes in a statistical distribution between two hypotheses. They observe the difference between the number of call setup requests (`INVITE` messages) and successfully complete handshakes (`200OK` reply messages). In normal traffic, these two types of messages should be equal at any given time. So when this ratio unexpectedly changes, it indicates a flooding attack. Reynolds and Ghosal describe a multi-layer detection scheme against DoS attack in VoIP [8]. They use a combination of sensors located across the network, continuously estimating the deviation from the long-term average of the number of call setup requests and successfully completed handshakes. Sengar *et al* present the VoIP Flooding Detection System (vFDS) for detecting anomalies in SIP traffic [9]. The Hellinger distance (HD) is used to measure abnormal deviation in VoIP packet streams. Traffic is divided into two sets and the dissimilarity between these sets is measured. The HD scheme has shown a strong flooding detection ability because low-rate flooding is likely to have a probability distribution that is different from that of normal traffic. However, an attacker can subvert this approach by only slightly increasing the attack traffic. Furthermore, it does not address how to maintain an accurate threshold during attacks as described in Hecht's work [10]. This problem increases the

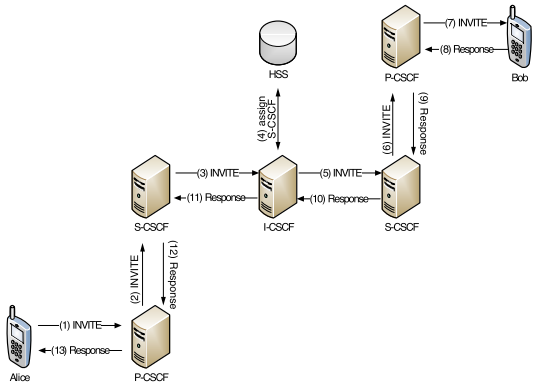



**Fig. 1** Basic session flow in IMS.

© 2014 Information Processing Society of Japan

likelihood of other attacks remaining undetected. Tang *et al* propose a similar approach to overcome the limitations of the previous schemes by using a sketch-based algorithm [11].

## 3. Flooding Detection Model

This section presents two statistical algorithms for detecting SIP flooding attacks. The first is an application of a Tanimoto distance which is used to measure a dissimilarity of selected SIP attributes. We use this algorithm because it is a plain computation method and can adapt to traffic changes, and therefore fits the dynamic environment of IMS. Second, we introduce an adaptive threshold of this system. In addition, we introduce a momentum oscillation indicator in order to detect a gradually increasing attack pattern. Base on a strong theoretical foundation, these algorithms can exhibit satisfactory performance over various attack types, without necessarily being complex or costly to implement.

### 3.1 Tanimoto Distance

In probability theory, a Tanimoto distance (TD) is used to measure the difference between two probability distributions [12]. To compute the TD, let $P$ and $Q$ be two probability distributions in the same sample space where $P$ and $Q$ are N-tuples $(p_1, p_2, \ldots, p_n)$ and $(q_1, q_2, \ldots, q_n)$. Then, the TD between $P$ and $Q$ is defined as

$$TD(P,Q) = \frac{\sum_{i=1}^{k} [\max(p_i, q_i) - \min(p_i, q_i)]}{\sum_{i=1}^{k} \max(p_i, q_i)}. \quad (1)$$

If the two probability distributions are totally different, TD approaches 1. This property provides a good approach to quantify the similarity of two data sets.

For our proposed flooding attack detection method, the traffic is divided into two portions: training and testing phases. The training phase, $P$, is assumed to be the probability distribution of a set of five SIP message types (`REGISTER`, `INVITE`, `200OK`, `ACK`, and `BYE`) as determined during a training phase of length $m*\Delta t$ timeslots. With the testing phase, $Q$, corresponds to the same probability distribution measured during a $\Delta t$ timeslot, shown in **Fig. 2**. Therefore, the 5-tuples of $P$ and $Q$ are $(p_{reg}, p_{int}, p_{ack}, p_{ok}, p_{bye})$ and $(q_{reg}, q_{int}, q_{ack}, q_{ok}, q_{bye})$ respectively where $p_{int}$ is the number of `INVITE` messages divided by the total number of five SIP message types in the training phase. The initial training phase is assumed to be free of any attacks and acts as a basis for comparison with the testing phase. Next, we measure the distance between these two phases, i.e., $TD_1$. A low TD value means there is no significant deviation between the two probability distributions. A high TD indicates that there are numerous open connections which are not closed in proper time. Then, it is implied that anomalies occurred in the traffic and altered the distributions.

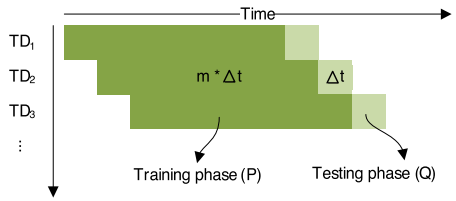



**Fig. 2** Sliding window mechanism of training phase and testing phase in the data stream.

After measuring the distance, if the distance does not exceed a threshold, a portion of the training phase will be merged with the testing phase to construct the next training phase, i.e., $TD_2$. This sliding window function helps the training phases to adapt to the dynamics of network traffic during a real-time analysis.

### 3.2 Adaptive Threshold

Most flooding attack detection methods face difficulty when determining the threshold for detection. It is hard to set an appropriate threshold value for a real time communication scenario. In particular, the IMS service traffic pattern will change over time. For example, calling traffic during the night may be less than during the day. A sudden increase in traffic can also occur, e.g., hot breaking news can cause a rapid increase in communication. These conditions are not necessarily caused by a DoS attack and need to be taken into consideration when setting a threshold. Therefore, a static threshold is neither practical nor responsive to expected normal traffic in this case. To accurately track normal traffic, we use an adaptive threshold in our system.

To deal with the fluctuation of IMS traffic, our adaptive threshold is based on an estimate of the mean deviation of the selected SIP packets computed from recent traffic measurements. In statistics, a moving average is widely used in time series analysis for predicting a future data set by using current and previous data sets. We preliminarily apply the Exponential Moving Average (EMA) for computing a distance threshold for the next time interval. Unlike the Simple Moving Average (SMA), EMA gives more weight to the latest data, which is suitable for IMS traffic environment. From Eq. (2), let $D_t$ and $D_{t-1}$ be estimated averages of the current and previous distances between two probability distributions: the training phase $P$ and testing phase $Q$. And $d_{t-1}$ is the previous distance. The coefficient $\alpha$ is a smoothing factor where $0 \leq \alpha \leq 1$. Using a small $\alpha$ we can detect small changes, and a larger value for detecting larger changes. Alternatively, $\alpha$ may be expressed in terms of $n$ time periods, where $\alpha = \frac{2}{n+1}$. For example, if one wants to calculate the EMA for the last 14 periods, $n$ is equal to 14.

$$D_t = \alpha d_{t-1} + (1-\alpha) D_{t-1} \quad (2)$$

Many anomaly traffic detection systems, such as Ref. [6], apply EMA as a threshold. However, because it utilizes only one single coefficient, EMA is not effective if there is a trend in the time series data [13]. Therefore, we add a trend forecast in the EMA as shown in Eq. (3)–Eq. (5). $\gamma$ is the trend smoothing factor where $0 \leq \gamma \leq 1$. Equation (3) adjusts $D_t$ directly for the trend of the previous period, $b_{t-1}$, by adding it to the last estimated distance value $D_{t-1}$. This helps to eliminate the lag and brings $D_t$ to the appropriate base of the current value. Equation (4) updates the trend, which is expressed as the difference between the last two values. This equation is similar to the basic form of EMA, but here applied to updating the trend. Note that there are several methods to choose the initial value of $b_1$, e.g., $b_1 = D_2 - D_1$. Finally, our adaptive threshold can be calculated by Eq. (5). We add $k$ times N-period standard deviation, $\sigma$, of the forecast values $(D_t + b_t)$ to reduce false alarms. The parameters $\alpha$, $\gamma$, and $k$ are used to set a safe margin for the threshold. We can tune proper

© 2014 Information Processing Society of Japan

values for them to achieve desirable detection accuracy.

$$D_t = \alpha d_{t-1} + (1-\alpha)(D_{t-1} + b_{t-1}) \quad (3)$$

$$b_t = \gamma(D_t - D_{t-1}) + (1-\gamma)b_{t-1} \quad (4)$$

$$TD_t^{\text{threshold}} = (D_t + b_t) + k\sigma \quad (5)$$

### 3.3 Momentum Oscillation Indicator

An attacker can potentially subvert an adaptive threshold if he knows the legitimate traffic intensity. In this scenario, during the first few periods, the attacker sends a very low intensity attack to the target server. This malicious traffic does not impact the threshold because there is no significant deviation in the traffic. Next, he increases the attack rate slightly which still does not affect the deviation of the overall traffic significantly. As the traffic increases, the adaptive threshold is updated. Finally, the attacker can send large malicious traffic that can block the IMS server without ever being detected because the traffic is below the current threshold. Thus, we propose a Momentum Oscillation Indicator (MOI) to detect such attack patterns.

To simplify the explanation of the calculation, this indicator divides the traffic intensity into two sides (upside and downside), by using the median over $n$ periods. The upside means the traffic intensity at that time is above the median while the downside means it is lower the median. The very first calculations for average upside and downside are simple $n$ period averages, as computed by Eq. (6) and Eq. (7). The second and subsequent calculations are based on the prior and the current upside and downside, as computed by Eq. (8) and Eq. (9). This is the same concept that EMA uses of comparing the prior value with the current value. Finally, in Eq. (10), the result is normalized and turned into an oscillator value that fluctuates between 0 and 100. The MOI is 100 when the average downside equals zero. This means the number of packets moved higher during all $n$ periods. There were no downsides to measure.

$$\text{Up}_1 = \frac{\sum_{i=t}^{t-n} \text{Up}_i}{n} \quad (6)$$

$$\text{Down}_1 = \frac{\sum_{i=t}^{t-n} \text{Down}_i}{n} \quad (7)$$

$$\text{avgUp}_t = \frac{(\text{Up}_{t-1} \times (n-1)) + \text{Up}_t}{n} \quad (8)$$

$$\text{avgDown}_t = \frac{(\text{Down}_{t-1} \times (n-1)) + \text{Down}_t}{n} \quad (9)$$

$$\text{MOI}_t = 100 - \frac{100}{1 - \frac{\text{avgUp}_t}{\text{avgDown}_t}} \quad (10)$$

## 4. Trust Model

According to NIST's technical report, anomaly detection systems are vulnerable to false positives [14]. DoS detection mechanisms, which aim at detecting floods, mainly look for sudden changes in the traffic and subsequently mark them as anomalous. However, they may produce false positives easily. The rationale behind detection methods is an assumption that the proportion between certain parameters remains roughly uniform as long as traffic is normal. A main drawback of detecting changes in the traffic volume is that the detection accuracy may be degraded if the legitimate volume is dynamic or suddenly increasing. This phenomenon happens easily and frequently in telecommunications such as during some flash events. For instance, cellphone networks were overwhelmed after the terror attacks in Boston [15]. The system cannot respond to all incoming requests. This traffic leads to significant changes in the distance between the current and previous traffic measurement. This causes false positives in any anomaly-based attack detection system. This is the major problem of a detection system because it causes a loss of confidence in the alerts. So we need a solution to confirm that a real attack is taking place before raising any alert. In general, a threshold tuning is the most widely-used method for false detection reduction. Increasing the threshold directly induces more false alarms, while many of them are actually not true. Reducing the threshold can reduce the number of false alarms, but such an action causes the detection to be unable to detect major attacks. This is the trade-off between reducing false alarms and maintaining system security.

In this work, we first address this problem by using a trust model. The trust model is integrated with the flooding attack detection algorithm to filter out a legitimate call after a deviation of traffic is detected. To calculate a reliable trust score, we use the call duration and its direction of each user to distinguish a legitimate user from a malicious user. This trust value can be used to construct the reliable social linkage with other users in the network through the trust inference mechanism. The social reliability, which is the evaluation of a user's behavior up to now, is also considered. A caller who conducts calling activities like a human will have a high trust value and a social reliability value. The system classifies this call as a legitimate call. If the average of trust score and social reliability of all callers in the testing phase is greater than the thresholds, the system will not raise the alarm even though the distance between the training and testing phases is high. Next, we will describe our trust calculation model in details.

### 4.1 Directed Trust Calculation

Trust has been traditionally proposed as a method to enhance security in many systems. The idea is to let parties rate each other and use the aggregated rating about a given party to derive a trust score, which can assist other parties in deciding whether or not to interact with that party in the future. In telecommunication systems, trust represents a model of past interactions between the calling party. We derive the effective directed trust calculation method from the previous work [16]. A trust value is automatically assigned to friends based on the outgoing call duration. The friend is a person who is already in a user's buddy list. The user can add or remove any call id of his/her friends in the list. Even though the system evaluates a trust value of each friend, a call from a friend is automatically forwarded to a callee. This trust value is used in the trust inference process when one does not have a relationship with this friend.

The original objective of this trust calculation over VoIP systems is for discriminating SPIT (Spam over Internet Telephony) calls from legitimate calls. Its effectiveness was already proved by comparing it with other techniques [16]. According to the

© 2014 Information Processing Society of Japan

technical report from 3GPP [17], a single SPITer can generate traffic around 250 GB per month. Besides of the huge traffic volume, generated by the SPITer and consuming network resources, the IMS server might be affected by the SPIT flooding. The similarity between a flooding attacker and a SPITer is they have very low social linkage with other legitimate users and their calling behaviors do not resemble human communication. Clearly, no legitimate user will make a call to a SPITer. In case of a flooding attacker, he may always use a new account to conduct a malicious activity. Therefore, our trust model can classify a malicious call through detecting these characteristics.

We assume that a user community in IMS system is represented as a social network. It contains nodes that are user equipment (UE). Every node maintains a buddy list. This social network is constructed by connecting a node to all the nodes in its buddy list. The buddy lists are kept in a central database on the provider side. Each buddy list is shared to others within the provider side during the trust computing process to protect user information. **Table 1** shows the data structure of a buddy list. The Friend attribute contains a call id of a friend. A call from a friend, who is already in a callee's buddy list, is automatically forwarded to a callee. The cumulative duration is the total outgoing call duration to this friend. Raw trust is the current trust, e.g., the trust that is computed in the current billing period. Since trust depends on past experiences with a person, a final trust is computed by combining a raw trust value with a historical trust value. The historical trust is the previous final trust of this friend. The range of a raw trust and a final trust are between 0 and 1. Please refer to Ref. [16] for the details of trust calculation.

### 4.2 Trust Inference

A trust value in the previous subsection is only assigned to friends in the buddy list who have direct interaction. In the real world, there is a possibility that a call will come from an unknown person. To estimate the trustworthiness of an unknown caller, we apply a situation found in daily human life. Generally, when encountering an unknown person, it is common for people to ask trusted friends for opinions about how much they can trust this new person. Therefore, we gather the trust values of an unknown caller from other friends in the network who already know about that person in order to classify a call. Hence, the property of trust in this work is transitivity. The trust values of friends in the buddy list will be shared to other nodes through a relationship path in the IMS network. These inferred trusts will be used when a caller and a callee do not have a direct relationship.

According to human reasoning, a person is much more likely to believe his friends as opposed a stranger. Similarly, a trusted acquaintance will also trust the beliefs of his friends. So, it is possible to find a path of friends from trustor to trustee with appropriate discounting [18]. Therefore, a multiplicative function is suitable for this case. The inferred trust is computed by Eq. (11), where $m$ is a user between a callee and a caller.

**Table 1** Data structure of a buddy list.

| Attribute | Data Type |
|---|---|
| Friend | CHAR |
| Cumulative duration | INTEGER |
| Raw trust | INTEGER |
| Final trust | INTEGER |

$$T_{callee,caller} = \prod_{m \in path}^{caller} T_{m,m+1} \tag{11}$$

We apply the seven degrees of separation phenomenon of a social network in the trust inference process. Everyone is connected through not more than seven intermediaries [19]. By this concept, we limit a relationship length between a callee and an unknown caller within a count of seven hops.

For newcomers or unknown callers of whom trust cannot be computed, the system will assign an initial trust value. The initial trust value is set to be slightly higher than a trust threshold. It is adjustable automatically after a user has calling activity. This initial trust assignment can eliminate the barrier for new users who do not have a trust value assigned by other users.

In a real network, there is a high possibility to have many trust paths between a caller and a callee. Reference [16] selects only one trust path that produces the highest trust value. However, selecting one trust path cannot reflect all the trust information of the caller. To compute the final inferred trust, we use the data fusion technique to combine the trust values of all trust paths together. Generally, data fusion is a process performed on multi-source data towards correlation, estimation and the combination of several data streams into one with a higher level of abstraction and greater meaningfulness. Its objective is to obtain an optimal decision or solution by combining many kinds of information from different sources. Next, we will explain the trust aggregation methodology.

#### 4.2.1 Dempster-Shafer Theory

In the field of statistics, the most well-known data fusion technique is the Bayesian theory:

$$P(H|E) = \frac{P(E|H)P(H)}{P(E)}. \tag{12}$$

Bayesian theory interprets a posterior probability, $P(H|E)$, as a measure of belief about a hypothesis or proposition $H$ updated in response to evidence $E$. The prior probability, $P(H)$, reflects the belief about $H$ in the absence of evidence. Researchers often estimate prior probabilities from empirical data, or, in the absence of empirical data, they assume them to be uniform or some other distribution. The outcome reflects these assumptions, so the critics of the Bayesian approach often point out that the method is not well-equipped to handle states of ignorance [20]. Clearly, this approach requires complete knowledge of both prior and conditional probabilities, which might be difficult to determine in practice. In contrast with the Bayesian approach, the Dempter-Shafer theory (DST) does not require the complete probabilistic model. We will now briefly introduce the key concepts of this theory. DST can be considered an extension of Bayesian inference [21]. It is a system for combining evidence from different sources and arrives at a degree of belief under uncertainty. Let a frame of discernment $\Theta = \{T, \neg T\}$ be two events under consideration; e.g., $T$ = trust in a caller, $\neg T$ = distrust in a caller.

Definition 1: Let $\Theta$ be a frame of discernment. A function $m : 2^{\Theta} \rightarrow [0, 1]$ is defined as Basic Belief Assignment (BBA)

© 2014 Information Processing Society of Japan

when it satisfies the following two properties:

$$m\{\emptyset\} = 0 \text{ and} \tag{13}$$

$$\sum_{A\subseteq\Theta} m(A) = 1. \tag{14}$$

Thus, we have $m(\{T\}) + m(\{\neg T\}) + m(\{T, \neg T\}) = 1$.

Definition 2: The belief function (*Bel*) for a set $A$ is defined as the sum of all the assignments of the subsets of $A$:

$$Bel(A) = \sum_{B\subseteq A} m(B). \tag{15}$$

For our case, we have

$$Bel(\{T\}) = m(\{T\}) \tag{16}$$

$$Bel(\{\neg T\}) = m(\{\neg T\}) \tag{17}$$

$$Bel(\{T, \neg T\}) = m(\{T\}) + m(\{\neg T\}) + m(\{T, \neg T\}). \tag{18}$$

For instance, suppose Alice and Bob are our friends who have a trust path to the unknown caller. Assuming Alice's trust path is trustworthy with a value of 0.8. Alice states that the caller is trustworthy. This means Alice's claim gives evidence for 0.8 degrees of belief in the caller's trustworthiness, but a zero degree of belief (not 0.2) that the caller is untrustworthy. This zero value mean that Alice's evidence gives no support to the belief that caller is untrustworthy. The 0.8 and the zero together constitute a belief function.

The combination of the evidence from different sources is done through the combination rule that is defined in the next definition.

Definition 3: Let $Bel_1$ and $Bel_2$ be belief functions over $\Theta$, with BBA $m_1$ and $m_2$. Then the function $m : 2^\Theta \to [0, 1]$ is defined by

$$m\{\emptyset\} = 0 \text{ and} \tag{19}$$

$$m(A) = \frac{\sum_{i,j:A_i\cap B_j=A} m_1(A_i)m_2(B_j)}{1-K}, \text{ where} \tag{20}$$

$$K = \sum_{i,j:A_i\cap B_j=\emptyset} m_1(A_i)m_2(B_j) \tag{21}$$

for all non-empty $A$.

Suppose that Bob's trust path is trustworthy with a value of 0.9, independently of Alice. Then we have

$$m_1(\{T\}) = 0.8, m_1(\{\neg T\}) = 0, m_1(\{T, \neg T\}) = 0.2$$

$$m_2(\{T\}) = 0.9, m_2(\{\neg T\}) = 0, m_2(\{T, \neg T\}) = 0.1$$

Finally, the aggregated trust value is

$$m_{12}(\{T\}) = 0.72 + 0.08 + 0.18 = 0.98.$$

#### 4.2.2 Social Reliability

The system will check the past behavior of a caller who interacts with other users to evaluate human interactivity. We call it the *social reliability* (SR) of a user. Two variables are considered to compute this feature: the degree of activity and the unique call.

*1) Degree of activity*: this is the ratio between incoming and outgoing calls during an observation period. Users who have a low degree of activity mean they make calls higher than the number of received calls. This can be the indicator for detecting a spammer. If the ratio is greater than 1, it is rounded down to 1. A high level of incoming calls might be a call center, who is not classified as a malicious user. A group of attackers can potentially mimic call activity by calling each other. However, on the telecommunication system, this attempt requires an extra cost.

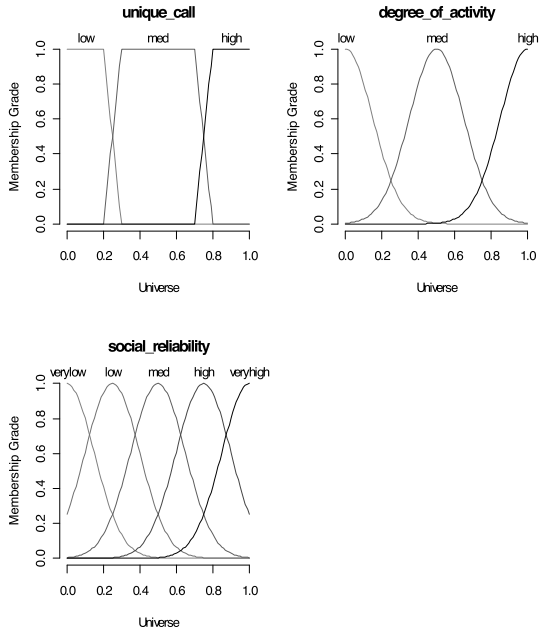



**Fig. 3** Fuzzy membership functions.

$$\text{degree of activity} = \frac{\text{incoming calls}}{\text{outgoing calls}} \tag{22}$$

*2) Unique call*: this feature allows us to identify an anomalous caller who makes a significant number of calls to different callees. Generally, a normal user will call a set of destination numbers, i.e., calls to callees that have already been contacted before.

$$\text{unique call} = \frac{\text{unique calls}}{\text{outgoing calls}} \tag{23}$$

Because these two variables are not the direct interaction between a caller and a callee, we will not use them as a trust value. Due to the imprecision of these values, we use fuzzy logic inference rules to calculate the social reliability of a caller. The degree of activity and the unique call are used as fuzzy descriptors. The membership function of these two variables is shown in **Fig. 3**. The example of the fuzzy inference rules are shown below.

- If the *degree of activity* is low and the *unique call* is low then the *social reliability* is verylow
- If the *degree of activity* is med and the *unique call* is low then the *social reliability* is low
- If the *degree of activity* is med and the *unique call* is med then the *social reliability* is med
- If the *degree of activity* is med and the *unique call* is high then the *social reliability* is high
- If the *degree of activity* is high and the *unique call* is high then the *social reliability* is veryhigh

This feature is used for filtering out a caller who has a low social interactivity with others. The SR of a user will be strong if he can balance his incoming and outgoing calls and most outgoing calls directly to a set of friends.

## 5. Evaluation

The accuracy of the flooding attack detection system is the ra-

© 2014 Information Processing Society of Japan

tio of correctly classified instances over the total number of instances. We focus on binary classification because it simplifies the analysis of a system's performance. As a binary classification problem, a classification may fall into one of the following four categories:

- True Positive (TP) - an actual attack triggers a detection system to produce an alarm
- False Positive (FP) - an event signaling a detection system to produce an alarm when no attack has taken place
- True Negative (TN) - no attack has taken place and no alarm is raised
- False Negative (FN) - a failure of a detection system to detect an actual attack

The performance metrics considered include the accuracy rate and the false positive rate (FPR):

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{24}$$

$$\text{FPR} = \frac{FP}{FP + TN}. \tag{25}$$

The accuracy rate is the proportion of true results, both TP and TN. FPR is the probability of falsely detecting a legitimate event as malicious. In addition, we investigate how the testing window size affects the detection of false negatives.

This section includes a description of the attack scenario, the system design, simulation information, and the performance evaluation of the detection system.

### 5.1 Attack Scenario

SIP flooding attacks can be divided into `REGISTER` flooding and `INVITE` flooding attacks. User Equipment (UE) needs to send `REGISTER` requests in order to register with the IMS server. In the case of `REGISTER` flooding, attackers send numerous bogus registration requests with invalid credentials in order to consume the processing resources of the server. The server will spend time looking into the database and sending back error messages which will be ignored by the attacker. For `INVITE` flooding, attackers send a large number of SIP `INVITE` messages to P-CSCF within a short period of time. As a transactional protocol, SIP requires the server to maintain a state for each `INVITE` message for some time period waiting for the associated `200OK` acknowledgement message. If the attack intensity is high enough, the resources of the server will be exhausted. We focus our evaluation on the `INVITE` flooding case first. Attacks utilizing other SIP attributes can be addressed in a similar way.

### 5.2 System Design

Under the assumption that most attacks come from outside the operator network, therefore, to analyze all incoming traffic, the detection system is located at the perimeter of the IMS as shown in **Fig. 4**. **Figure 5** shows the overview of the detection system. Before starting the detection system, the number of timeslots $m$ and the duration $\Delta T$ of the training and testing phases must be defined. Increasing the time span of the training and testing phases will increase the number of packets in the analysis. Since our proposed system quantifies the correlations among SIP attributes, the number of packets during a learning process does not highly

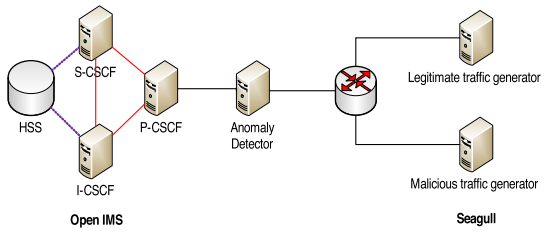



**Fig. 4** Testbed topology.

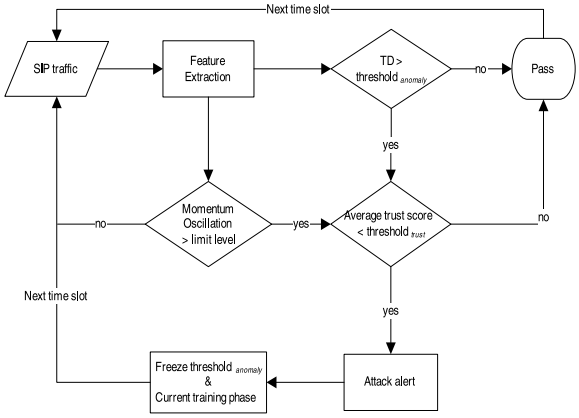



**Fig. 5** System overview.

affect the detection performance. However, there is a trade-off between a longer and a shorter time span because of the response time of each packet. A longer time span can accurately return distance values, while shorter time spans can detect small changes. Assuming the first few training phases are free from any malicious traffic. The incoming SIP traffic is extracted into five SIP message types. The distance between the training and testing phases are computed by TD and then compared with the adaptive threshold. If the distance does not exceed the threshold, the next training phase and testing phase are continuously evaluated, and so on. However, when a flooding attack comes, it will disturb the probability distribution obtained from the testing phase. Thus the distance will exceed its threshold. When this happens, the system will check the trust values and the SR values of all callers in the testing phase. These two variables are used to discriminate a legitimate caller from a malicious caller. Concurrently, the momentum oscillation of the traffic is also monitored. If it is higher than the desired level over the predefined time period, the trust and SR will be analyzed. The attack alarm will be raised when the average trust and SR value of the callers are less than the threshold. Then, the system will keep the current training phases and anomaly threshold, and only move the testing set to the next time interval. As a result, the distance in the next cycle is evaluated between the stored training phase and the current testing phase. This freezing process will keep on until the distance drops below the anomaly threshold. This can protect the threshold from being impacted by the flooding attacks and makes it stable during attack. The system will resume the normal process when the detected anomaly is no longer present. Consequently, the traffic during attack periods will never be included in the training phase. The traffic sampling technique is not applied in this work because we aim at analyzing all incoming traffic to the server and proposing a near real-time detection system. Since the lightweight statisti-

© 2014 Information Processing Society of Japan

cal calculation is applied to analyze the data, the system does not require high computing resources. In contrast with other methods that require sampling the data, they need to keep the whole traffic in the memory before analyzing. This requires more computing resources and cannot be executed in real-time. An alarm notifies the system administrator about a current attack attempt and anomalous activity. The system will generate a report that provides key information which can be used to identify the attack's origin such as caller id and attack time. This is helpful for the administrators to take the next action regarding this suspicious event. To protect the entire system, we need to incorporate our proposed system with other protecting systems or countermeasure mechanisms. In a practical deployment, it can be transparently interposed at a firewall and implemented as a loadable module of the firewall. This is out of a scope of this work.

### 5.3 Dataset

Due to privacy concerns, we have not found any publicly available IMS traffic dataset. Therefore, we use Seagull [22], the IMS traffic generator, to synthetically generate the traffic. Seagull can generate SIP messages and has the ability to simulate customized SIP scenarios. So we used Seagull to generate both legitimate traffic and attack traffic. We used the Open IMS Core [23] to emulate an IMS system. It is an implementation of IMS CSCFs and a lightweight HSS, which together form the core elements of all IMS/NGN architectures. The four components are based on Open Source software (e.g., the SIP Express Router (SER) and MySQL). Many researchers simulated their testbed by only generating a set of users and randomly choosing call parties, such as Refs. [10] and [11]. In order to simulate the real conditions of the IMS user network and make our testbed close to the real environment, we use the communication between users in the Wikipedia talk network as our IMS users [24]. **Table 2** shows the characteristics of this social network. The network contains all the users and discussion from the inception of Wikipedia until January 2008. We select this dataset because it contains the communication direction among users and large enough for the evaluation. Initially, this dataset does not have positive and negative links among them. We assume that the nodes who have direct communication are friends in a buddy list. The path length between users does not affect the trust model as proved by the previous work [16].

The SIP session initiation of a legitimate call was generated synthetically with a Poisson distribution, which was the same as the PSTN model [25]. The call duration times are heavy-tailed distributions that can be generated with Pareto distribution [25]. The mean number of calls per unit of time and call duration are defined according to the IP phone statistics from the operator [26]. These statistics were collected from April 1st 2011 to March 31, 2012. According to this data, call duration ranged between 104 and 215 seconds. The average call frequency in this study is around 200 calls per second. The call destination is selected either from a buddy list or another person who is not in a buddy list. The choice of this recipient is a Zipfian distribution [27].

**Table 2** Wikipedia talk network characteristics.

| | |
|---|---|
| Node | 2,394,385 |
| Edges | 5,021,410 |
| Average clustering coefficient | 0.1958 |
| Diameter (longest shortest path) | 9 |

The attacks are generated synthetically. This allowed us to control the characteristics of the attacks, and hence be able to investigate the performance of the detection algorithms for different attack types. Our experiment considered both high and low intensity attacks, whose rates varied from 100 to 400 calls per second. This range was chosen to evaluate the effectiveness of our detection approach under both low rate and high rate attacks. The duration of the attacks was normally distributed with a mean of 60 seconds. We consider attacks whose intensity increases both abruptly and gradually.

The window size of a training phases and a testing phases are set by $\Delta t$. According to the SIP specification [28], an `INVITE` transaction timeout is 32 seconds. Consequently, to correlate an `INVITE` message with a response message, the sampling window size should be 32 seconds. However, our proposed method is not sensitive to per-flow information. Therefore, we set sampling size ($\Delta t$) equal to 10 seconds in order to achieve a high detection accuracy. Moreover, the distance measurement algorithm depends on the training phase window size, $m * \Delta t$. A longer training phase accurately returns the distance value, while shorter training phases can detect a small change. Then, we set the training phase to 40 seconds ($4*10$) in order to balance the responsiveness and the detection accuracy. Other parameters are set as $\alpha = 0.2$, $\gamma = 0.2$, $k = 2$, and the standard deviation is calculated from the last 20 time intervals. The parameter $n$ of the MOI is 20. This value can be lowered to increase sensitivity or raised to decrease sensitivity. The trust threshold and SR threshold are 0.25. These are the optimal values in our environment.

Next, we will describe four experiments that show the detection accuracy of the system and prove that the trust model can reduce false positives effectively.

### 5.4 The Detection without the Trust Model

This experiment investigates the performance of the detection system without the trust model integration under multiple attack intensities. The mean amplitude of the low intensity attack in this experiment was 50% of the legitimate traffic mean rate. The attack traffic was injected every five minutes starting from the 598th second with a low rate, 100 calls per second. The attack rates of the next three floodings were 200, 300, and 400 calls per second respectively. **Figure 6** shows the results for the TDs and their adaptive thresholds. The horizontal axis started from the 50th second according to the length of the first training and testing phases. The learning period ended at the 250th second and then started calculating a threshold. The MOI rate during the attacks is shown in **Fig. 7**. In this work, we consider the traffic as behaving in an anomalous way when MOI is above 80% (red dash line). This level can be adjusted to better fit the traffic environment. When the MOI is above the desired level, the previous median before an attack is stored and used as the median for the current state. This median value is kept until MOI falls below the limit level again. The alarm is raised when the momentum is greater

© 2014 Information Processing Society of Japan

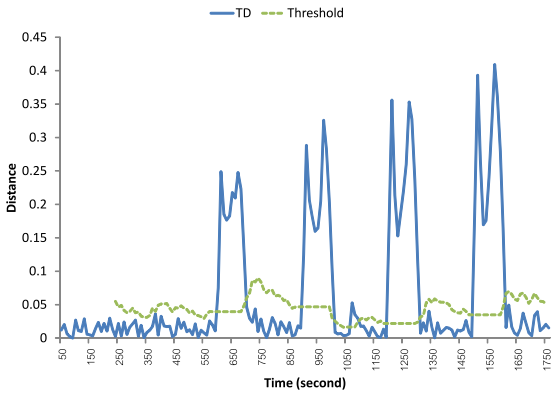



**Fig. 6** TDs and their adaptive thresholds during four flooding attack rates: 100-400 calls/sec.

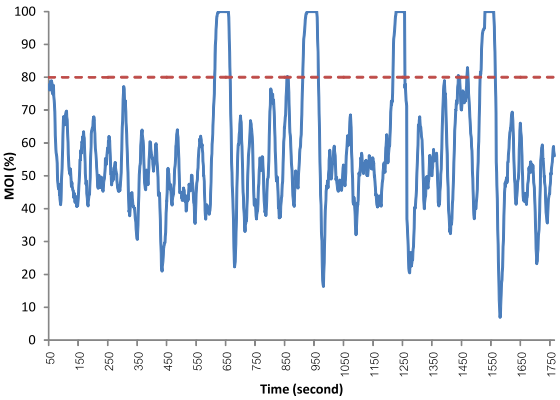



**Fig. 7** MOI during during four flooding attack rates: 100-400 calls/sec.

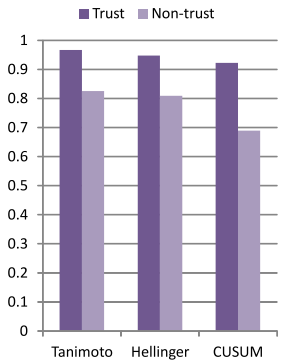



**Fig. 8** The accuracy rate between trust and non-trust approaches.

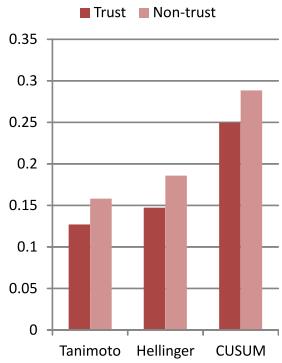



**Fig. 9** False positive rate between trust and non-trust approaches.

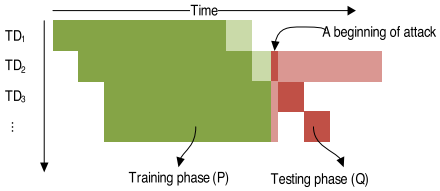



**Fig. 10** A FN may occur in a longer testing phase size.

than the desired level over the predefined time period. The first attack was detected at the 600th second and at the 614th second by the TD and MOI respectively. The highest attack intensity, around 400 calls per second, was injected at the 1,498th second. This attack traffic is 200% of a legitimate traffic. The distance significantly deviated at the 1,500th second. From these results, our system produced a very small detection delay. Moreover, according to these graphs, both low and high intensity attacks were detected accurately. The sensitivity of the detection system in this experiment was 100%. However, false positives occur when legitimate traffic suddenly increased, e.g., at the 1060th second in the Fig. 6. We reduce this false alarm by using the trust model so that its performance will be shown in the next experiment.

### 5.5 False Positives Reduction

We conducted this experiment to investigate the performance of detection after integrating the trust model. We compared our proposed system with two well-known anomaly detection algorithms: Hellinger distance (HD) [9] and Cumulative Sum (CUSUM) [6]. **Figure 8** and **Fig. 9** show the average accuracy rate and FPR between trust and non-trust integration approaches, respectively. After integrating the trust model, the average detection accuracy of TD, HD, and CUSUM algorithms were increased by around 14.17%, 13.87%, and 23.3% respectively. You can also see that the trust model integration method can reduce the FP in all flooding attack detection algorithms. According to Eq. (24), since false positives were reduced, the accuracy rate increases. It indicates that the trust model can classify a legitimate call correctly. Moreover, from these results, our proposed system produced the best detection performance compared to other methods in term of the highest accuracy and the lowest FPR.

### 5.6 Testing Window Size

The above results were for specific values of the testing phase window size. Next we investigate the trade-off between the window size and the detection accuracy. Generally, in anomaly-based detection, a longer testing phase is set to obtain a stable distribution under normal conditions. However, an attack detection probability will be reduced because of an FN. This FN occurs when a few portions of the testing phase contain an initial part of attack traffic, as shown in **Fig. 10**. If this attack portion is not high enough to alter the distribution, then no attack alarm is raised. If the testing phase is set to be short, many FPs will be raised. This is the general issue of a flooding attack detection system that uses training and testing phases. However, with our trust integration method, we can reduce FN while increasing detection accuracy by resizing the testing phase. **Figure 11** shows the FN and accuracy rate among different testing window sizes of our detection system. FN were reduced when the testing phase window size was decreased. With our trust integration, FP were also reduced even though the size was decreasing. Consequently, the accuracy

© 2014 Information Processing Society of Japan

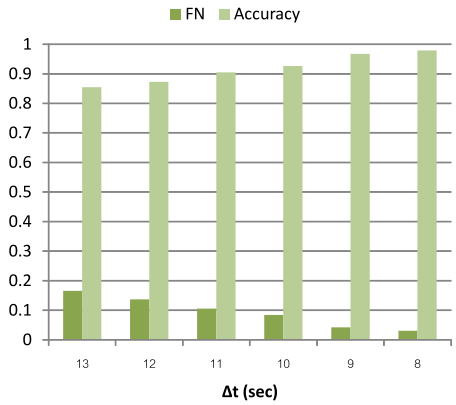



**Fig. 11** FN and accuracy rate among different testing phase sizes.

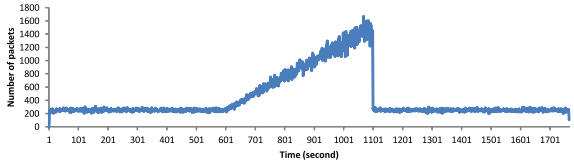



**Fig. 12** SIP traffic during a gradually increasing flooding attack.

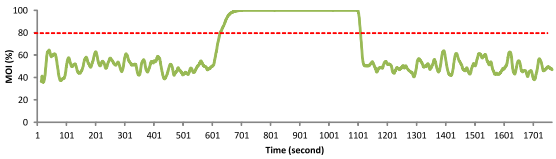



**Fig. 13** MOI of the traffic.

rate of the detection was improved. However, reducing a testing window size will consume more computing resources. Therefore, we suggest using the appropriate size that fits the operator's policies and resources.

### 5.7 A Gradually Increasing Attack Pattern

This experiment investigates the performance of MOI for detecting a gradually increasing attack detection. **Figure 12** shows the SIP traffic including a gradually increasing flooding attack. The attack traffic was injected at time 601st second with an attack rate of 1% of the normal traffic and then increased slightly until it reaches 500% of the normal traffic at time 1100th second. This kind of attack can subvert an adaptive threshold technique, as explained in Section 3.3. However, as shown in **Fig. 13**, the proposed MOI can accurately detect it. The MOI increased rapidly after time 601st second and then reached the highest level at 678th second. The MOI remained 100% until the SIP traffic intensity was decreased at time 1101st second. This result shows that the MOI can detect such attack pattern correctly.

## 6. Discussion

Generally, a trust-based system will be subverted if an attacker has a high trust value assigned by other nodes. Since the duration of an outgoing call is used to compute a trust value, an attacker needs some calls from other users frequently to maintain high trust values. At the same time, the attacker would have to maintain a balance in his in-out calling degrees in order to keep an appropriate social reliability level. In the VoIP system, this activity requires extra cost that would make it counterproductive for the attacker's business. However, for the proposed model to be efficient, an attacker must not be able to steal the identity of a legitimate user. The trust model will be useless if an attacker can use the trust score of a legitimate user. This is the authentication issue of the IMS that is out of the scope of this work. However, there are many works that have been proposed to fix this problem such as in Ref. [29], where an Identity Based Cryptography (IBC) is employed to enhance the security of the IMS authentication process.

Another concern is the performance comparison with a modern application layer session-based firewall. This firewall keeps track of the network connection sessions and holds significant attributes of each session in memory. Then, CPU and memory resources are required for analyzing the session. Generally, in order to prevent the memory from filling up, sessions will time out if no traffic has passed for a certain period. These state connections are removed from the memory. However, during an attack, more and more requests come in, which have to be processed by the firewall. As a consequence, the firewall cannot monitor every packet, so some of them are not recognized by the security system or must be dropped because of a lack of buffer capacity. Comparing with our stateless detection system, we apply the statistical method to detect an anomaly of traffic. The system does not require a large memory to store a processed data. This can avoid a bottleneck problem when the detection system needs to process massive traffic.

## 7. Conclusion and Future Work

Along with the accelerated global deployment of IMS networks, their security problem has become increasingly serious. Undoubtedly, Denial of Service attack presents a serious threat to IMS networks. In this paper, we proposed an anomaly-based DoS attack detection system using the Tanimoto distance, an adaptive threshold, and a momentum oscillation indicator. These algorithms are stateless and require low computation overhead. The detection system extracts SIP traffic messages and estimates the dissimilarity between them over predefined time period. Because of the correlation of the chosen packets, an attacker needs to mimic the legitimate traffic through complete SIP transaction in order to subvert our system. The modified moving average is computed as an adaptive threshold for tracking the behavior of the traffic and making the system more accurate. A momentum oscillator indicator is proposed to detect a special attack pattern, a gradually increasing attack. Furthermore, we address the false alarm problem by using the trust model. The trust value is calculated from the call behavior of each user. Performance evaluation on the testbed simulation showed that our detection system successfully detected various flooding attack patterns. The average accuracy rate was higher than 90%. Furthermore, the false positives were reduced after using the trust model. Lastly, the experimental results showed that decreasing a testing phase's window size can improve the detection performance while simultaneously reducing the false negatives.

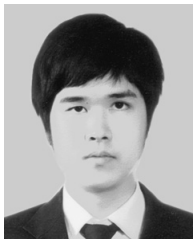
In this paper, we have used SIP traffic as the monitoring factor for flooding detection. So, in the future work, we will study how to monitor other resources for attack detection. Moreover,

© 2014 Information Processing Society of Japan

the system has a few numeric parameters that influence its performance. We intend to investigate how the value of these parameters can be automatically determined. We are also aware of the limitations of the IMS datasets used. We aim to evaluate the performance of our detection system on real IMS data.

## References


[1] Poikselka, M. and Mayer, G.: *The IMS: IP Multimedia Concepts and Services*, Wiley, United Kingdom, 3rd edition (2009).
[2] 3GPP: Security Architecture, TS 33.102 V10 (2011).
[3] 3GPP: Access Security for IP-based Services, TS 33.203 V10.2 (2011).
[4] Varghese, G., Fingerhut, J.A. and Bonomi, F.: Detecting evasion attacks at high speeds without reassembly, *SIGCOMM Computer Communication Review*, Vol.36, No.4, pp.327–338 (2006).
[5] 3GPP: IP Multimedia Subsystem (IMS), TS 23.228 V11.7 (2013).
[6] Siris, V. and Papagalou, F.: Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks, *Global Telecommunications Conference*, Dallas, TX, pp.2050–2054 (2004).
[7] Rebahi, Y., Sher, M. and Magedanz, T.: Detecting Flooding Attacks against IP Multimedia Subsystem (IMS) Networks, *IEEE/ACS International Conference on Computer Systems and Applications*, Qatar, pp.848–851 (2008).
[8] Reynolds, B. and Ghosal, D.: Secure IP Telephony using Multilayered Protection, *10th Annual Network and Distributed System Security Symposium*, San Diego, CA (2003).
[9] Sengar, H., Wijesekera, D. and Jajodia, S.: Detecting VoIP Floods Using the Hellinger Distance, *IEEE Trans. Parallel and Distributed Systems*, Vol.19, No.6, pp.794–805 (2008).
[10] Hecht, C., Reichl, P., Berger, A., Jung, O. and Gojmerac, I.: Intrusion Detection in IMS: Experiences with a Hellinger Distance-Based Flooding Detector, *1st International Conference on Evolving Internet*, France, pp.65–70 (2009).
[11] Tang, J., Cheng, Y. and Zhou, C.: Sketch-Based SIP Flooding Detection Using Hellinger Distance, *IEEE Global Telecommunications Conference*, pp.1–6 (2009).
[12] Duda, R., Hart, P. and Stork, D.: *Pattern Classification*, 2nd edition, Wiley, United States (2001).
[13] NIST/SEMATECH: e-Handbook of Statistical Methods (2003).
[14] Mell, P.: An Overview of issues in testing intrusion detection systems, Technical Report, NIST (2003).
[15] Farrell, M.: Cellphone networks overwhelmed after blasts in Boston, available from ⟨http://b.globe.com/14uWliO⟩ (accessed 2013-08-14).
[16] Chaisamran, N., Okuda, T. and Yamaguchi, S.: Trust-based VoIP Spam Detection based on Calling Behaviors and Human Relationships, *Journal of Information Processing*, Vol.21, No.2, pp.188–197 (2013).
[17] 3GPP: Study of mechanisms for Protection against Unsolicited Communication for IMS, TS 33.937 V11 (2012).
[18] Guha, R., Kumar, R., Raghavan, P. and Tomkins, A.: Propagation of trust and distrust, *13th International Conference on World Wide Web*, pp.403–412 (2004).
[19] Leskovec, J. and Horvitz, E.: Planetary-scale views on a large instant-messaging network, *17th International Conference on World Wide Web*, pp.915–924 (2008).
[20] Chen, T. and Venkataramanan, V.: Dempster-Shafer theory for intrusion detection in ad hoc networks, *IEEE Internet Computing*, Vol.9, No.6, pp.35–41 (2005).
[21] Shafer, G.: *A Mathematical Theory of Evidence*, Princeton University Press (1976).
[22] Seagull, available from ⟨http://gull.sourceforge.net⟩ (accessed 2013-04-25).
[23] FOKUS: Open IMS Core, available from ⟨http://www.openimscore.org⟩ (accessed 2013-04-25).
[24] Leskovec, J., Huttenlocher, D. and Kleinberg, J.: Predicting Positive and Negative Links in Online Social Networks, *19th International Conference on World Wide Web*, pp.641–650 (2010).
[25] Dang, T., Sonkoly, B. and Molnar, S.: Fractal Analysis and Modeling of VoIP Traffic, *11th International Telecommunications Network Strategy and Planning Symposium*, pp.123–130 (2004).
[26] NTT: IP Phone Usage Statistics and Network Information 2011 (Japanese), available from ⟨http://www.ntt-east.co.jp/info-st/network/traffic_h23/⟩ (accessed 2013-04-25).
[27] Balasubramaniyan, V., Ahamad, M. and Park, H.: CallRank: Combating SPIT Using Call Duration, Social Networks and Global Reputation, *4th Conference on Email and Anti-Spam* (2007).
[28] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E.: SIP: Session Initiation Protocol, RFC3261 (2002).
[29] Abid, M., Song, S., Moustafa, H. and Afifi, H.: Efficient Identity-based Authentication for IMS Based Services Access, *7th International Conference on Advances in Mobile Computing and Multimedia*, pp.260–266 (2009).

**Noppawat Chaisamran** received his B.Sc. degree in ICT from Mahidol University, Thailand, and M.E. degree in Information Science from Nara Institute of Science and Technology (NAIST), Japan in 2007 and 2011 respectively. He is currently a Ph.D. student in the Graduate School of Information Science, NAIST. His research interests are in the area of IP telephony and telecommunication security.



**Takeshi Okuda** received his M.E. and D.E. degrees in Information Science from Osaka University, Japan in 1998 and 2011 respectively. He is currently an Associate Professor in the Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Japan. His research interests include virtual machine, virtual network and their security. He is a member of IEEE.



**Youki Kadobayashi** received his Ph.D. degree in Computer Science from Osaka University, Japan, in 1997. He is currently an Associate Professor in the Graduate School of Information Science, Nara Institute of Science and Technology (NAIST), Japan. Since 2013, he has also been working as a Rapporteur of ITU-T Q.4/17 for cybersecurity standardization. His research interest includes cybersecurity, web application security, and distributed systems. He is a member of ACM, IEICE, IEEE Communications Society, and IPSJ.

© 2014 Information Processing Society of Japan



**Suguru Yamaguchi** received his M.E. and D.E. degrees in Computer Science from Osaka University, Japan, in 1988 and 1991, respectively. Since 2000, he has been a Professor of the Graduate School of Information Science, Nara Institute of Science and Technology (NAIST). He has been working aggressively for making and running JPCERT/CC since 1996, and APCERT since 2002. From 2004 to 2010, he was appointed to Advisor on Information Security to the Cabinet, Government of Japan. His research interests include technologies for information sharing, multimedia communication over broadband channels, large-scale distributed computing systems including cloud computing technology, network security and network management for the Internet.

© 2014 Information Processing Society of Japan