Detection of SIP Flooding Attacks based on the Upper Bound of the Possible Number of SIP Messages

Jea-Tek Ryu, Byeong-Hee Roh and K i-Yeol Ryu
Graduate School of Information and Communication, Ajou University, Suwon 443-749, South Korea
[e-mail: {ricman, bhroh, kryu}@ajou.ac.kr]
*Corresponding author: Byeong-Hee Roh

Received July 10, 2009; revised August 26, 2009; accepted September 10, 2009; published October 30, 2009

Abstract

Since SIP uses a text-based message format and is open to the public Internet, it provides a number of potential opportunities for Denial of Service (DoS) attacks in a similar manner to most Internet applications. In this paper, we propose an effective detection method for SIP flooding attacks in order to deal with the problems of conventional schemes. We derive the upper bound of the possible number of SIP messages, considering not only the network congestion status but also the different properties of individual SIP messages such as INVITE, BYE and CANCEL. The proposed method can be easily extended to detect flooding attacks by other SIP messages.

Keywords: SIP flooding, flooding detection, SIP flooding attack, SIP security, SIP threat

This research was supported by the MKE (The Ministry of Knowledge Economy), the Korean government, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency (NIPA-2009-C1090-0902-0003).

DOI: 10.3837/tiis.2009.05.006

1. Introduction

Session Initiation Protocol (SIP) [1] has been adopted as the main signaling and session management protocol for most recent multimedia applications and systems such as Voice over IP (VoIP) or IP Multimedia Subsystem (IMS). Since SIP has a text-based message format and is open to the public Internet, it provides a number of potential opportunities for Denial of Service (DoS) attacks similar to most Internet applications. Generally, DoS attacks are aimed at denying or degrading a legitimate user's access to a service or network resources, or bringing down the servers offering such services [2]. For a DoS attack, attackers may generate a massive number of malicious SIP request messages to a target SIP server in order to bring down the server. In most research, such an attack is called an SIP flooding attack.

There has been much work on dealing with DoS attacks on SIP services [3]. Most of this work has been focused on signaling DoS attacks on individual SIP signaling messages such as INVITE, BYE, or CANCEL. Since signaling DoS attacks have a unique pattern for each attack, encryption-based approaches have been largely used in most research [4][5][6]. On the other hand, there has not been much related research for SIP flooding attacks based on cumulative sum (CUSUM) [7], Hellinger Distance (HD) [8] and adaptive threshold [9]. Comparative analysis for the three algorithms is given in [10]. These methods have two major weak points. First, they detect SIP flooding attacks by investigating whether the number of incoming SIP messages exceeds a certain threshold level. Generally, the threshold value is given statistically using a priory-analyzed normal SIP message pattern when no congestion is assumed. However, when network congestion occurs, the number of incoming SIP messages may increase because of the retransmission nature of SIP messages. Accordingly, as the degree of network congestion increases, the number of incoming SIP messages exceeds the threshold, and the previous methods may identify this behavior as a flooding attack even when it is still a normal condition, i.e. no attack is occurring. Second, the previous methods mainly focused on the INVITE message flooding attack. However, flooding attacks are caused by other SIP messages such as BYE, CANCEL and other SIP messages [11]. Applying the detection methods that are suitable for INVITE flooding attacks to detect the BYE and CANCEL flooding attacks is inefficient, due to the different properties of those messages.

In this paper, we propose an effective detection method of SIP flooding attacks in order to deal with the problems of the conventional schemes. We derive the upper bound of the possible number of SIP messages, considering not only the network congestion status but also the different properties of the individual SIP messages such as INVITE, BYE and CANCEL. The proposed method can be easily extended to detect flooding attacks by other SIP messages.

The rest of the paper is organized as follows: Section 2 briefly describes some SIP issues related to the proposed method. Section 3 presents the algorithm for determining the upper bound of the possible number of SIP messages, considering network congestion. Section 4 describes the proposed algorithms to detect the INVITE, BYE, and CANCEL flooding attacks. Section 5 gives the experimental results of the proposed schemes. Section 6 gives the extension of the proposed methods and Section 7 concludes the paper.

2. Background

SIP [1] is an application layer protocol which enables multimedia sessions or calls to be set up, maintained, modified or terminated. SIP messages are text-based. It is similar to that of HTTP and is used in request and response message pairs. SIP messages consist of a header which includes signaling information, and a body which provides additional information such as audio and video codecs for session setup.

2.1 SIP Signaling Procedure

2.1.1 Session Establishment

The SIP session setup is a three-way handshake involving INVITE, 200 OK, and ACK as shown in **Fig. 1**. The SIP user agent (UA) transmits an INVITE request message to the receiver UA in order to create a session via the SIP proxy servers. The proxy servers receive and forward the INVITE message without disrupting it. When the receiver UA receives the INVITE message, it sends a 200 OK message to accept the session request. Then, the sender UA confirms the session setup by sending the ACK message, and session setup is completed.

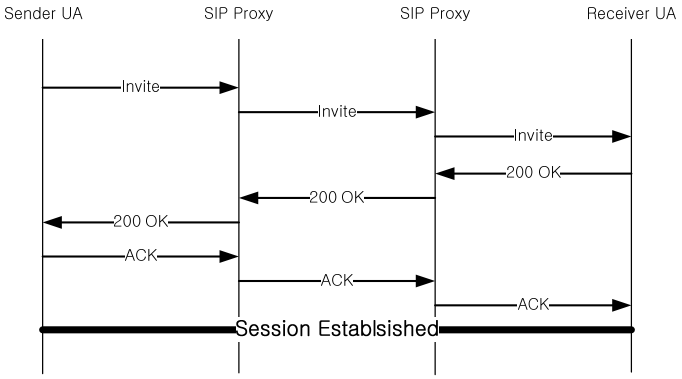


Fig. 1. Session establishment via an INVITE message

2.1.2 Session Cancellation

Session cancellation occurs when the sender UA wants to end the session setup process prior to setup completion, or the receiver UA denies the INVITE request, or there is no response to the request from its corresponding application. As an example, session cancellation initiated by the receiver UA is depicted in **Fig. 2**. The receiver UA transmits a CANCEL request message to the sender UA, and each SIP element replies with a 200 OK message to accept the CANCEL.

2.1.3 Session Termination

Session termination occurs when either UA sends a BYE message referencing an existing session that was successfully established using the INVITE/200OK/ACK exchange. The session termination process is shown in **Fig. 3**. When a UA requests session termination by sending a BYE message, the other UA responds with the 200 OK message. Then the session is terminated. It is noted that INVITE and BYE are end-to-end methods while CANCEL is a hop-by-hop request. That is, a proxy receiving a CANCEL request immediately responds with

a 200 OK response. On the other hand, the 200 OK response for the INVITE or BYE requests is initiated by the UA receiving the request.

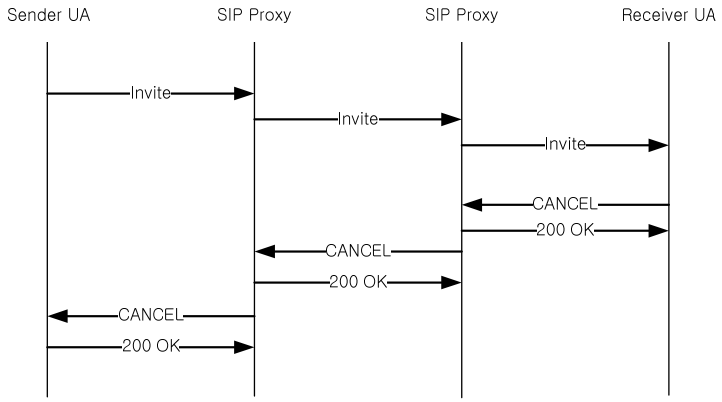


Fig. 2. Session cancellation via a CANCEL message

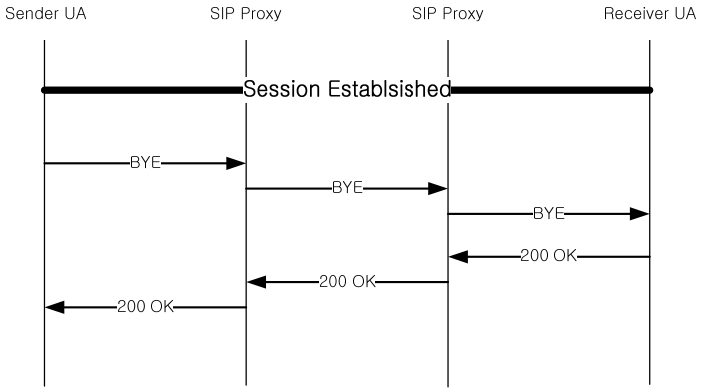


Fig. 3. Session termination via a BYE message

2.2 SIP Retransmission Mechanism

SIP messages are generally sent over UDP. Therefore, packet loss by network congestion may occur. For reliable message transmission, there needs to be a retransmission mechanism to deal with this. In RFC 3261[1], two retransmission mechanisms of SIP messages for the INVITE and non-INVITE requests are defined as follows.

2.2.1 Retransmission of INVITE request messages

If there was no response from the receiver in T1 (the default is 500ms) after transmission of the INVITE message, the sender UA assumes that the message was lost in the network. Then, the sender UA retransmits the message and every retransmission doubles the previous retransmission interval until message transmission succeeds. Retransmission occurs until time 64*T1 (the default is 32sec) from initial message transmission. It means the maximum number of retransmissions cannot exceed seven including the initial transmission.

3. Upper Bound of the Possible Number of SIP Messages

SIP messages are generally sent over UDP. If a SIP element cannot receive a response for its requested message due to data loss or delay, it retransmits the message in the application layer, as explained in the previous Section. As we can easily imagine, the frequency of retransmissions increases in proportion to the degree of network congestion. In other words, the higher the degree of network congestion, the more SIP messages that an SIP element receives due to retransmission.

As we mentioned earlier, CUSUM or HD-based existing methods [7][8][9][10] detect SIP flooding attacks, especially INVITE flooding attacks, based on a pre-determined constant or a slightly changed threshold value derived from the statistics on normal traffic behavior, without considering network congestion. This means that their detection may be falsely triggered by a higher network congestion status. Users should set a higher threshold value to overcome the problem, but this can lead to longer detection times or failure of detection entirely. In addition, the existing methods can only be applied to INVITE flooding. However, flooding attacks are caused by other SIP messages such as BYE and CANCEL. Applying the detection methods that are suitable for INVITE flooding attacks to the BYE and CANCEL flooding attacks is inefficient, due to the different properties of those messages, as explained in Section 2.

In this section, we derive the upper bound of the possible number of SIP messages, considering the degree of network congestion status. Then, the detection methods for the INVITE, BYE and CANCEL flooding attacks using the upper bound will be described in the next section. It is assumed that the SIP elements such as the SIP proxy servers can know the degree of network congestion via a network core such as a network management system (NMS). It is usually given as p, the retransmission rate due to network congestion. The assumption is reasonable because IMS and QoS-enabled VoIP services with SIP proxy servers are usually managed by network service operators with their own NMSs. It is also assumed that the time is divided into a constant period of Δ, which is a basic unit for traffic measurement. Let t_n and a_n, n=0,1,2,…, be the n-th time period and the number of newly generated SIP messages received at a SIP element during t_n respectively. Then, the following lemmas give the upper bounds of the possible number of SIP messages for several retransmission mechanisms including those explained in Section 2.2.

Lemma 1. Upper Bound for Infinite Retransmission Mechanism

For the infinite retransmission mechanism, unresponded messages are repeatedly retransmitted until successful responses are observed. Let A be the maximum number of SIP messages newly generated and received (other than by retransmission) by an SIP element. Then, the upper bound of the possible number of messages that the SIP element can receive during a given time period is given by

$$M_U^1 = A\frac{1}{1-p} \tag{1}$$

where p denotes the retransmission rate due to network congestion.

Proof. During t_0, only newly generated SIP messages, a_0, are sent to the SIP element. Among those SIP messages, $p \cdot a_0$ messages are not responded to due to network congestion

and are retransmitted during the next period t_1. Let m_k (k=0,1,2,…) be the total number of SIP messages generated during t_k. Then we can obtain $m_1 = a_1 + p \cdot a_0$. Generally,

$$m_k = a_k + p \cdot a_{k-1} = \sum_{i=0}^{k} p^{k-i} \cdot a_i , \qquad \text{k=0,1,2,…} \tag{2}$$

Since $a_k \le A$ for $\forall k$ and $0 \le p \le 1$, we can obtain the upper bound of the possible number of messages that the SIP element can receive during a given time period as follows:

$$M_U^1 \equiv \lim_{k\to\infty} m_k \le \lim_{k\to\infty} \sum_{i=0}^{k} A \cdot p^{k-i} = A\frac{1}{1-p} \tag{3}$$

It is noted that A is used as a threshold value to determine whether it is an attack or not for the CUSUM or HD-based conventional schemes. As we can see from Eq. (3), the threshold value should be increased as the degree of network congestion becomes higher.

Lemma 2. Upper Bound for INVITE Retransmission Mechanism

The INVITE retransmission mechanism is described in Section 2.2.1. Let $RT_{\max}^{INVITE}$ be the maximum number of retransmissions. Then, the upper bound of the possible number of SIP messages during a given time period is given by

$$M_U^2 = A \sum_{i=0}^{RT_{\max}^{INVITE}} p^i \tag{4}$$

Proof. Let r_k (k=1,2,…, $RT_{\max}^{INVITE}$) be the time period that the k-th retransmission for an unresponded message is done after its initial transmission. It is noted that $r_k = \sum_{i=0}^{k-1} 2^i r_i$ where $r_0 = T_1$. Then, the total number of SIP messages during t_k is given by

$$m_k = a_k + \sum_{i=1}^{RT_{\max}^{INVITE}} a_{k-r_i} p^i \tag{5}$$

Since $a_k \le A$ for $\forall k$, we have the same upper bound as in Eq.(4).

Lemma 3. Upper Bound for non-INVITE Retransmission Mechanism

The non-INVITE retransmission mechanism is described in Section 2.2.2. Let $RT_{\max}^{non-INVITE}$ be the maximum number of retransmissions. Then, we can obtain the upper bound of the possible number of SIP messages during a given time period for the non-INVITE retransmission mechanism as follows

$$M_U^3 = A \sum_{i=0}^{RT_{\max}^{non-INVITE}} p^i \tag{6}$$

Proof. We can easily derive Eq.(6) in a manner similar to Lemma 2.

It is noted that the default values of $RT_{\max}^{INVITE}$ and $RT_{\max}^{non-INVITE}$ are 6 and 10 respectively, as described in Section 2.Fig. 4 shows the three upper bounds, M_U^1, M_U^2 and M_U^3 obtained by varying p when A is 50. As we can see, as p increases, the upper bounds also increase. And, the upper bounds show almost the same values when p is less than 0.5, which represents general environments in a well-controlled network. Therefore, we can simply use Eq. (1) for the upper bound of the possible number of SIP messages under a given p.

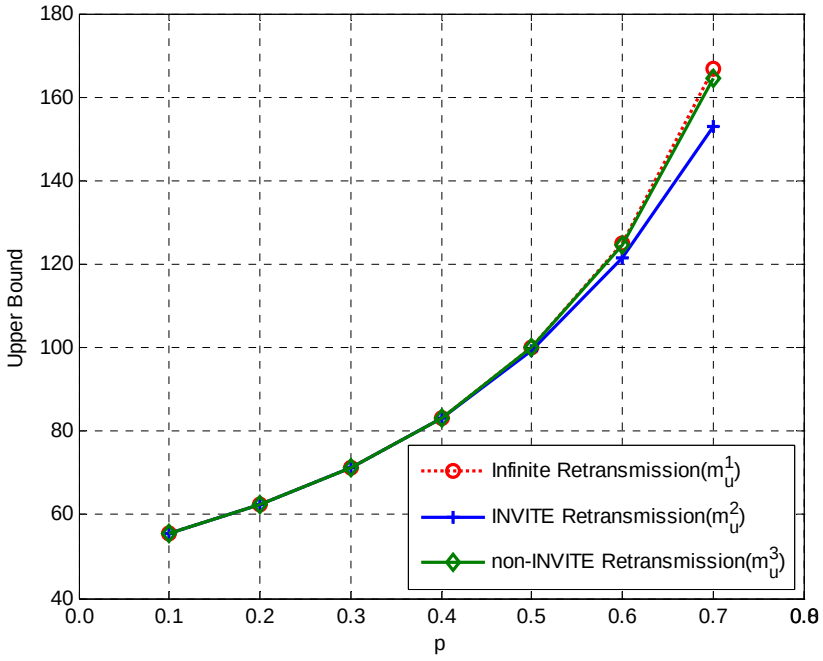


Fig. 4. Upper bound variation (A=50)

4. Detection of SIP Flooding Attacks

As we mentioned before, existing methods to detect SIP flooding attacks [7][8][9][10] focused on the INVITE flooding attacks. In this section, we explain effective methods to detect SIP flooding attacks by not only INVITE but also CANCEL and BYE. As will be discussed in Section 6, the proposed method can be easily extended to detect various flooding attacks with SIP messages other than those of INVITE, CANCEL and BYE. For the detection method, the upper bound explained in the previous section is used.

4.1 Algorithm 1: Detection of INVITE Flooding Attacks

A SIP element, e.g. a SIP proxy server, can receive INVITE messages from any possible SIP sender UAs. Let R_k be the weighted average of m_k, which is the total number of INVITE messages received at the SIP element during the k-th time period t_k. Then we can obtain

$$R_k = \alpha_1 \cdot R_{k-1} + (1-\alpha_1) \cdot m_k \text{ , k=0,1,2,…} \quad (7)$$

where α_1 is a constant value for the weighted average between 0 and 1.

As we can easily imagine, the normal INVITE traffic flow in a stationary state varies within a forecastable range when no re-transmission is considered. Therefore, a threshold value A for the normal INVITE message flow can be determined based on actual measurement prior to the detection process by the administrative policy of the network operators. It is noted that the threshold value should be modified according to the network congestion status, as shown in Section 3. However, since existing methods use a fixed threshold value, they cannot reflect the situation.

In order to reflect the network congestion status and determine the situations in which INVITE flooding attacks occur, we define the three states of NORMAL, ALERT, and ATTACK, as in [12]. The NORMAL state is where there are no attacks. The ALERT state is where an attack may have occurred but this has not yet been completely decided. In the ATTACK state, it is inferred that the SIP element has been attacked. The state transitions are shown in **Fig. 5**.

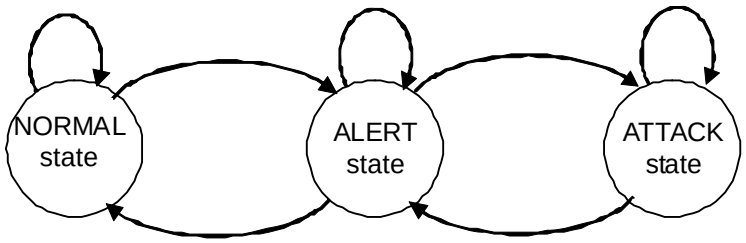


Fig. 5. State transitions for attack detection

Let p_k be the retransmission rate due to network congestion during the k-th time period t_k. This means that the retransmission rate is dynamically changeable according to the network conditions. With R_k and p_k, we can detect the INVITE flooding attacks according to the following algorithm, as shown in **Fig.6**.

<variables>

count : counter for representing the degree of attack

$C_{\max}$: maximum of count

L_{ALERT} : threshold of count for ALERT state

L_{ATTACK} : threshold of count for ATTACK state

state : current state of the algorithm

<main algorithm>

Initially, k=0, count=0, state=NORMAL.

```
At each time period t_k ,
    - update the weighted average R_k with measured m_k using Eq.(7)
    - it is assumed that p_k and A_k are known
if (state==NORMAL)
    if ( R_k > A_k /(1 − p_k ) )
    count++;
    else
    count = MAX(0, count--);
    endif
    if ( count > L_ALERT )
    state=ALERT
    endif
elseif (state==ALERT)
    if ( R_k > A_k /(1 − p_k ) )
    count++;
    else
    count --;
    endif
    if ( count ≤ L_ALERT )
    state=NORMAL
    elseif (count > L_ATTACK )
    state=ATTACK
    endif
else
    if ( R_k > A_k /(1 − p_k ) )
    count = MIN ( C_max , count++);
    else
    count --;
    endif
    if (count ≤ L_ATTACK )
    state=ALERT;
    endif
endif
```

Fig. 6. Algorithm 1: INVITE Flooding Attack Detection

4.2 Algorithm 2: Detection of BYE Flooding Attacks

As we mentioned before, any SIP UA can send INVITE messages at any time. However, BYE messages can only be sent from UAs which have previously established sessions that have not been terminated. Accordingly, in case of the BYE message, in order to obtain the upper bound, only the UAs with established sessions are considered.

Let N_k be the number of SIP sessions in progress at the end of the k-th time period. The SIP proxy servers can easily calculate N_k by monitoring the three-way handshake of INVITE, 200 OK, and ACK, as shown in **Fig. 1**, and the message exchange for session termination, as shown in **Fig. 3**. That is, $N_k = N_{k-1} + G_k - T_k$ where G_k and T_k are the respective number of newly established and terminated sessions during that time(t_k). Let L_{BYE} be the number of

successive time periods to get $A_{\mathrm{BYE}}(k)$, which is the maximum number of sessions in progress during the L_{BYE} time periods before the k-th time period, and is defined as

$$A_{\mathrm{BYE}}(k) \equiv \max(N_{k-L_{\mathrm{BYE}}+1}, \cdots, N_{k-1}, N_k), \quad \text{k=0,1,2,…} \tag{8}$$

We use the maximum N_k's during L_{BYE} in order to reflect the nature of dynamic fluctuation of SIP session establishments and terminations. Let $R_{\mathrm{BYE}}(k)$ be the weighted average of $A_{\mathrm{BYE}}(k)$, which is given by

$$R_{\mathrm{BYE}}(k) = \alpha_2 \cdot R_{\mathrm{BYE}}(k-1) + (1-\alpha_2) \cdot A_{\mathrm{BYE}}(k), \quad \text{k=0,1,2,…} \tag{9}$$

where α_2 is the constant value for the weighted average between 0 and 1. Using the parameters defined above, the detection algorithm of BYE flooding attacks is given in **Fig. 7**. As shown in **Fig. 7**, the algorithm considers the worst case where all the established sessions are trying to terminate with a given retransmission probability during the time period. It is noted that only UAs with ongoing sessions are eligible to terminate by sending BYE messages. Accordingly, the situation where the number of BYE messages that the proxy server receives during a given time period exceeds the upper bound of the worst case is caused only by BYE flooding attacks.

<variables>
count : counter for representing the degree of attack
$C_{BYE,\max}$: maximum of count
$L_{BYE,ALERT}$: threshold of count for ALERT state
$L_{BYE,ATTACK}$: threshold of count for ATTACK state
state : current state of the algorithm

<main algorithm>
Initially, k=0, count=0, state=NORMAL.
At the end of each time period t_k,
- obtain N_k the number of sessions in progress during t_k
- calculate $A_{\mathrm{BYE}}(k)$ and $R_{\mathrm{BYE}}(k)$
- it is assumed that p_k and A_k are known.

```
if (state==NORMAL)
    if ( R_BYE(k) > A_BYE(k)/(1 − p_k) )
    count++;
    else
    count = MAX(0, count--);
    endif
    if ( count > L_BYE,ALERT )
    state=ALERT
    endif
elseif (state==ALERT)
    if ( R_BYE(k) > A_BYE(k)/(1 − p_k) )
```

```
        count++;
        else
        count --;
        endif
        if ( count ≤ L_BYE,ALERT )
        state=NORMAL
        elseif (count > L_BYE,ATTACK )
        state=ATTACK
        endif
    else
      if ( R_BYE(k) > A_BYE(k)/(1 − p_k) )
        count = MIN ( C_BYE,max , count++);
        else
        count --;
        endif
        if (count ≤ L_BYE,ATTACK )
        state=ALERT;
        endif
    endif
```

Fig. 7. Algorithm 2: BYE Flooding Attack Detection

4.3 Algorithm 3: Detection of CANCEL Flooding Attacks

The SIP CANCEL method is used for pending session cancellation. As shown in **Fig. 2**, CANCEL messages can be sent from UAs trying to set up sessions which have not yet been established. It is noted that BYE messages can be sent from UAs that have previously set up sessions. Let N_k^C be the number of SIP sessions to be set up but not yet established at the end of the k-th time period. Let L_C be the number of successive time periods to get $A_{\text{CANCEL}}(k)$, which is defined below in a manner similar to Eq.(8)

$$A_{\text{CANCEL}}(k) \equiv \max(N_{k-L_C+1}^C, \cdots, N_{k-1}^C, N_k^C), \quad \text{k=0,1,2,…} \tag{10}$$

Let $R_{\text{CANCLE}}(k)$ be the weighted average of $A_{\text{CANCEL}}(k)$, which is given by

$$R_{\text{CANCEL}}(k) = \alpha_3 \cdot R_{\text{CANCEL}}(k-1) + (1-\alpha_3) \cdot A_{\text{CANCEL}}(k), \quad \text{k=0,1,2,…} \tag{11}$$

where α_3 is the constant value for the weighted average between 0 and 1.

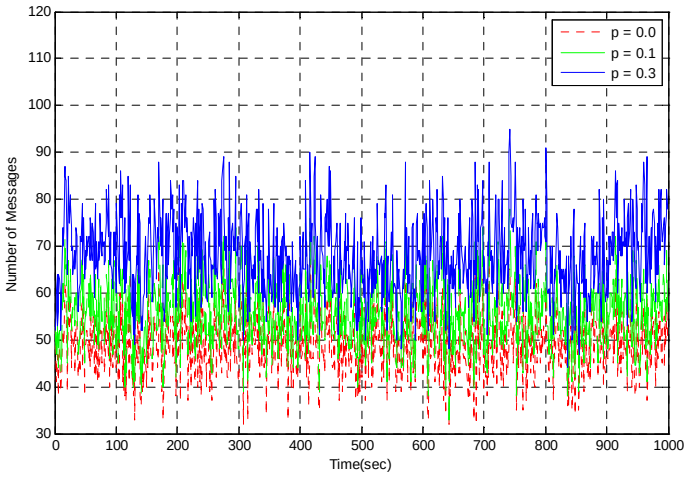
Using the parameters defined above, we can obtain a detection algorithm of CANCEL flooding attacks similar to **Fig. 7**. The entire process of detection is the same as the BYE flooding attack detection algorithm, except for the following differences: there are CANCEL specific parameters such as $C_{C,\max}$, $L_{C,ALERT}$ and $L_{C,ATTACK}$, which mean the maximum and respective threshold count values for the ALERT and ATTACK states. The condition for incrementing the count is given by $R_{\text{CANCEL}}(k) > A_{\text{CANCEL}}(k)/(1-p_k)$ for N_k^C.

5. Experimental Results

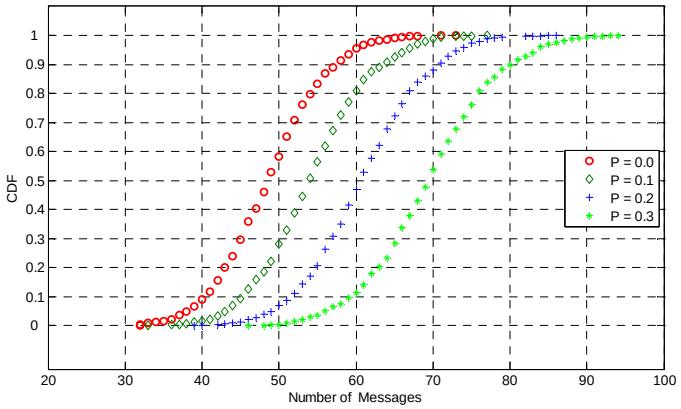
5.1 Effect of Retransmission Rate

In order to show the effect of the retransmission rate (p) on the number of generated SIP messages, we wrote an SIP message generation program which generates SIP messages according to an exponentially distributed inter-arrival time where the average number of messages per time period (M) varies with respect to p. In order to simplify the experimental results, we used the following terms. Let $s(t_n; M, p)$ be the number of generated SIP messages during time period t_n given M and p. Let $s_{\text{mean}}(M, p)$ and $s_{\max}(M, p)$ be the mean and maximum values of the sequence $\{s(t_n; M, p) \mid t_n \geq 0\}$, respectively.

Fig. 8(a) and **Fig. 8(b)** show the sequences of $\{s(t_n; M, p) \mid t_n \geq 0\}$ and their cumulative density functions (CDFs), respectively, when M is fixed at 50 and p varies from 0 to 0.3. As is shown in **Fig. 8**, as the retransmission rate increases, the number of SIP messages also increases in proportion to about $1/(1-p)$ times the average rate.



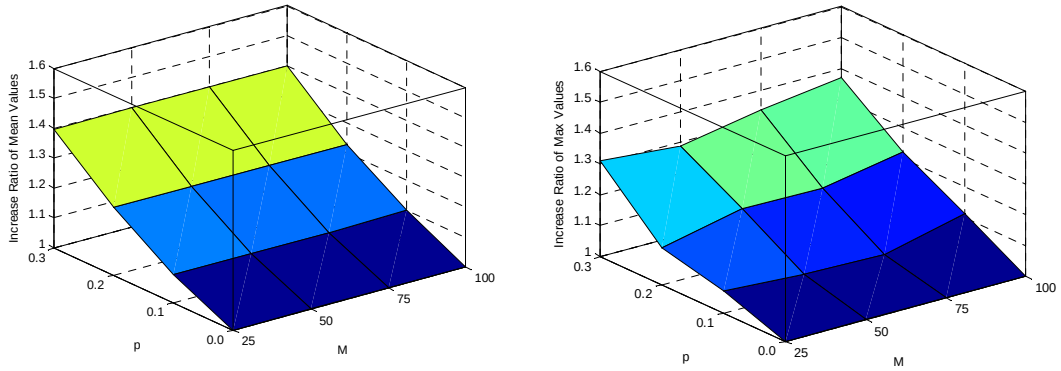
(a) Example Sequence of the Number of Generated SIP Messages



(b) CDF of the Sequence

Fig. 8. Effect of Retransmission Rate on the Number of SIP Messages (M=50)

We also carried out an experiment to show how $s_{\text{mean}}(M, p)$ and $s_{\max}(M, p)$ are affected by the variation of M and p. **Fig. 9(a)** shows the ratio of the mean number of SIP messages when p varies from 0.0 to 0.3 i.e., for p=0.0, $s_{\text{mean}}(M, p)/s_{\text{mean}}(M,0.0)$. And, **Fig. 9(b)** depicts $s_{\max}(M, p)/s_{\max}(M,0.0)$. We can see that both the mean and maximum number of SIP messages increase as p and M increase.



(a) the ratio of $s_{\text{mean}}(M, p)/s_{\text{mean}}(M,0.0)$ (b) the ratio of $s_{\max}(M, p)/s_{\max}(M,0.0)$

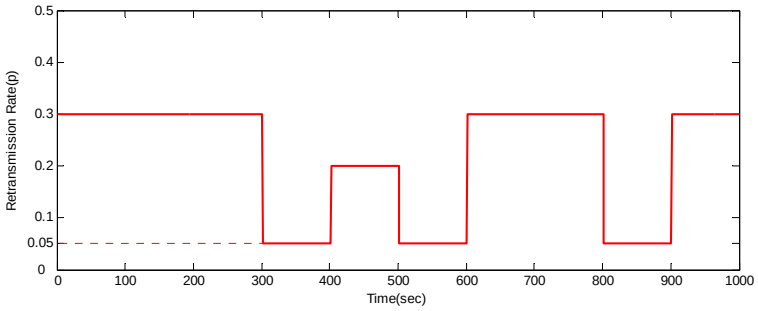
Fig. 9. Increment Effect on the Number of SIP Messages due to Variation of M and p

5.2 Effectiveness of Algorithm 1

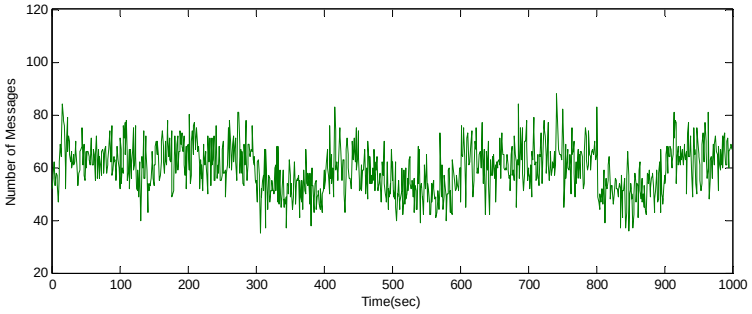
In order to evaluate the effectiveness of Algorithm 1 to detect INVITE flooding attacks, we constructed a normal sequence for the number of INVITE messages, as shown in **Fig. 10(b)**, from the sequence with p=0.0 shown in **Fig. 8(a)**, where p varied according to the pattern shown in **Fig. 10(a)**. We also made INVITE flooding attack sequences during a certain period, as shown in **Fig. 10(c)**.

In **Fig. 11**, the performance of INVITE flooding attack detection using our proposed Algorithm 1 is illustrated. For the experiment with Algorithm 1, we used the following parameters: α_1 =0.5, $C_{\max}$ =6, L_{ALERT} = 1, and L_{ATTACK} = 5. From **Fig. 11(a)**, we can see that the variation of the upper bound for the normal INVITE sequence reflects that of p shown in **Fig. 10(b)** exactly. **Fig. 11(b)** and **Fig. 11(c)** depict the attack count and state calculated by Algorithm 1, respectively. The interval where the state shows ATTACK indicates that the INVITE flooding attack has definitely been detected and we can see that the ATTACK duration is similar to the interval of the INVITE flooding attack, as shown in **Fig. 10(c)**.

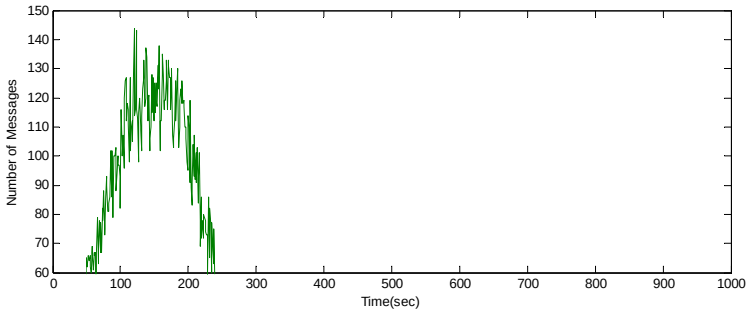
Fig. 12 shows the results of the CUSUM-based method proposed in [7] to detect INVITE flooding attacks for the sequences shown in **Fig. 10**. In **Fig. 12**, the time points where CUSUM values exceed the threshold are classified as occurrences of INVITE flooding attacks. While the CUSUM-based method can easily detect the normal state when p is very low, i.e., near to 0.0, it shows false positives in the time interval where p is larger than 0.1. That is, parts of time intervals where p is larger than 0.1 are classified as attacks, even though no attacks occur during those intervals. It is expected that this kind of false positive may be derived by other methods not considering the degree of network congestion.



(a) Variation of p

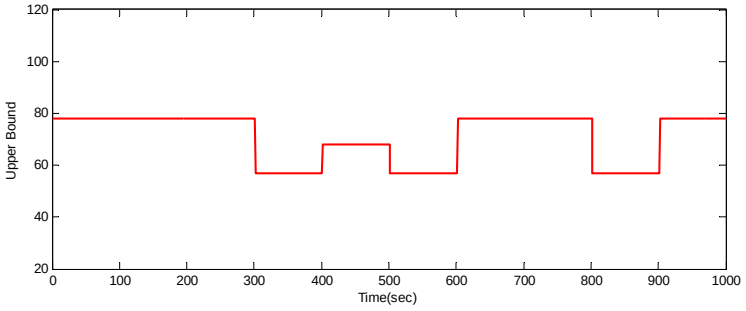


(b) Normal INVITE sequence reflecting the variation of p as in (a)

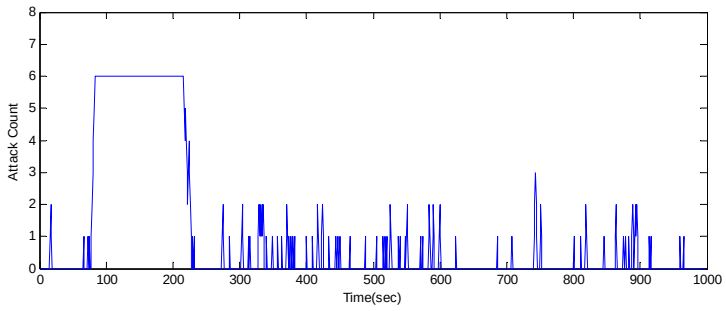


(c) INVITE flooding attack sequence

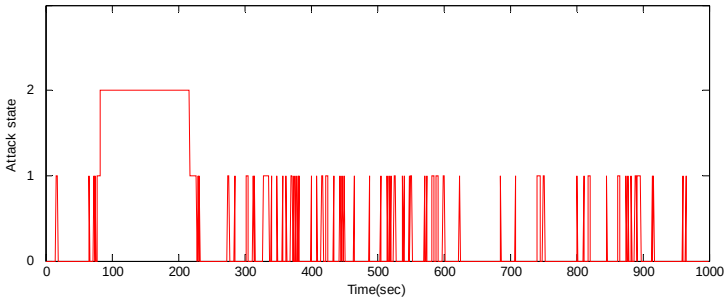
Fig. 10. Normal and Attack INVITE Sequences for Algorithm 1



(a) Upper bound for the normal sequence of **Fig. 10(b)**



(b) Attack count



(c) Decision state (state 0:NORMAL, 1:ALERT, 2:ATTACK)

Fig. 11. Attack detection by Algorithm 1

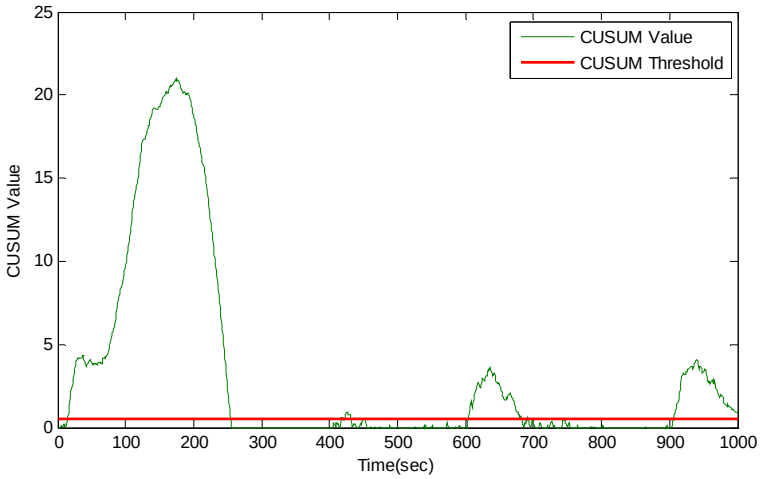
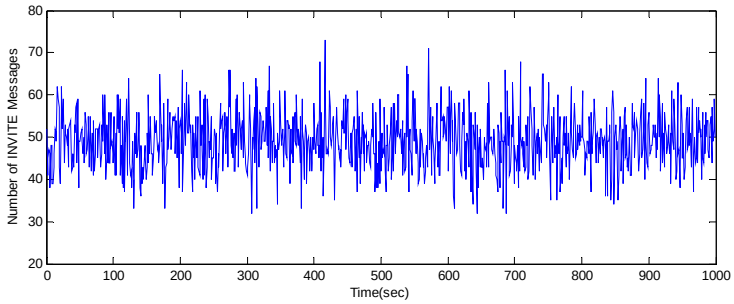


Fig. 12. Attack detection by CUSUM [7]

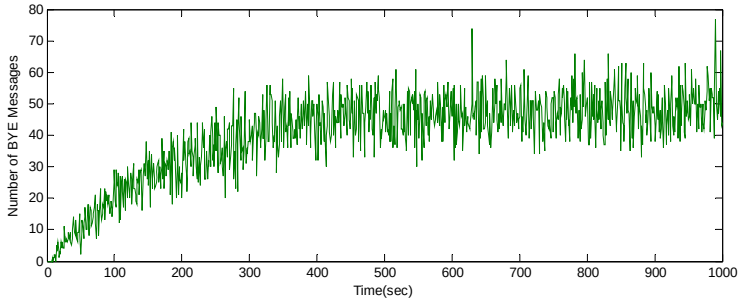
5.3 Effectiveness of Algorithm 2

In order to evaluate the effectiveness of Algorithm 2, we carried out the following experiment. First, we used the normal INVITE message sequence, as shown in **Fig. 13(a)**, which is the same sequence as shown in **Fig. 8(a)** when p=0.0. It is assumed that sessions for all INVITE messages generated during a given time period are successfully established within this period, and each session has an exponentially distributed lifetime with a mean time of 180 seconds.

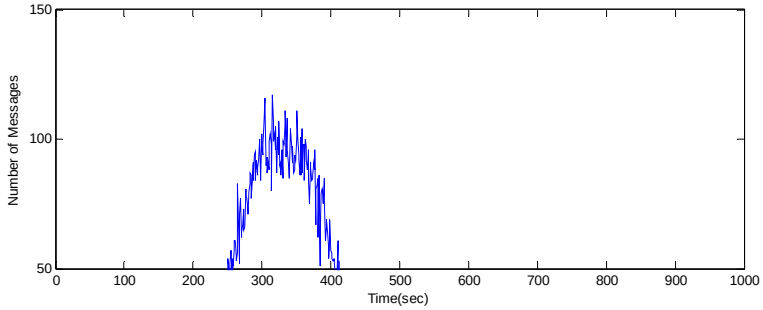
Fig. 13(b) shows the number of normal BYE messages to terminate corresponding sessions when p=0.0. Then, we generated a BYE flooding attack sequence, as shown in **Fig. 13(c)**.



(a) INVITE messages



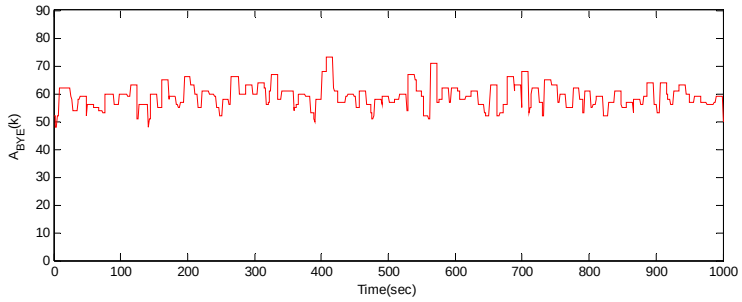
(b) Normal BYE message Sequence (when p=0.0)



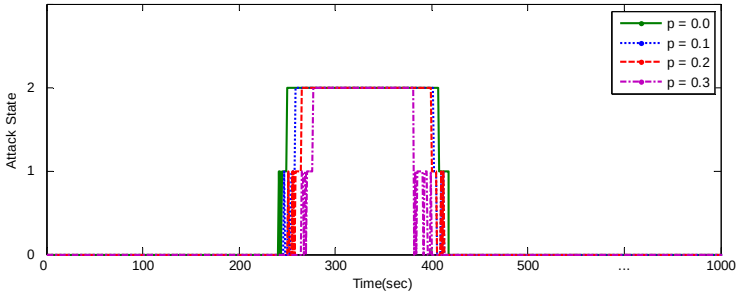
(c) BYE Flooding Attack Sequence

Fig. 13. Sequences used for the experiment with Algorithm 2

Fig. 14(a) shows the sequence of the maximum number of maintained SIP sessions during the time window defined in Section 3.2. The detection results of Algorithm 2 when p varies are shown in **Fig. 14(b)**. It is shown that the time intervals of the BYE flooding attack detected by Algorithm 2 are very close to the actual attack interval shown in **Fig. 13(c)**. However, as p increases, the detected attack interval decreases compared to the actual attack interval. This means that the attack decision time becomes longer. However, we can see that the differences are very small. So, it can be said that Algorithm 2 is effective for detection of BYE flooding attacks.



(a) Sequence of $A_{\mathrm{BYE}}(k)$

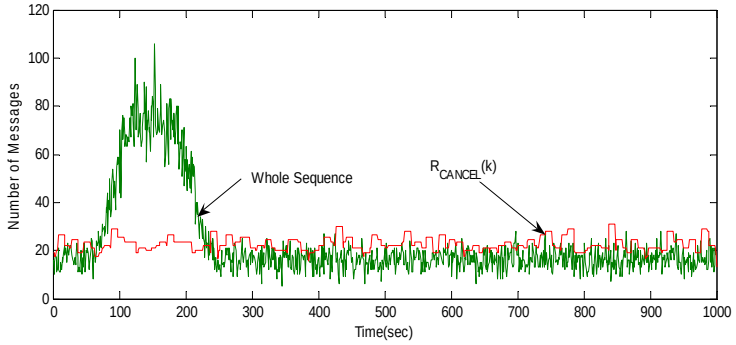


(b) Attack state (state 0:NORMAL, 1:ALERT, 2:ATTACK)

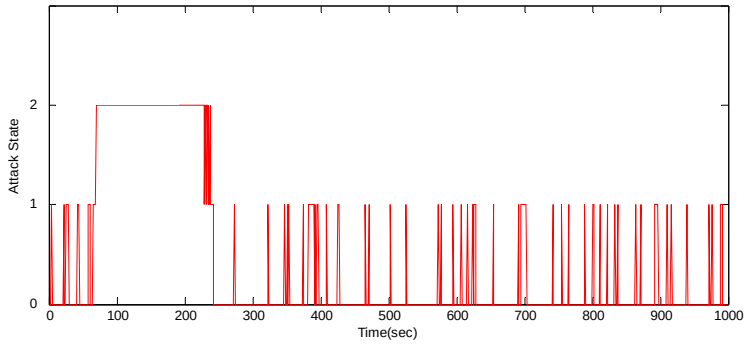
Fig. 14. Attack detection result by Algorithm 2

5.4 Effectiveness of Algorithm 3

In order to show the performance of Algorithm 3 to detect CANCEL flooding attacks, we carried out the following experiment. First, we used the same normal INVITE message sequence as shown in **Fig. 13(a)**. We assumed that some of those INVITE messages are cancelled by sending CANCEL messages. Let r be the ratio of cancellations among INVITE requests. The worst case of cancellation is where r is 1, in which all the INVITE requests will be cancelled during the next time period. Then, the sequence shown in **Fig. 10(c)** was used for CANCEL flooding attacks. Fig. 15(a) shows the sequence that consisted of both normal and attack CANCEL messages and the values of $R_{\mathrm{CANCLE}}(k)$ used for Algorithm 3. For the normal CANCEL messages, r is set to 0.3. That is, 30% of all INVITE requests are cancelled.



(a) Sequences of entire messages and $R_{\mathrm{CANCEL}}(k)$



(b) Attack state (state 0:NORMAL, 1:ALERT, 2:ATTACK)

Fig. 15. CANCEL Flooding Attack Detection by Algorithm 3

The detected interval of the CANCEL flooding attack is shown in **Fig. 15(b)**, which is very close to the actual attack interval shown in **Fig. 10(c)**.

6. Discussion

In addition to INVITE, CANCEL, and BYE, there are other sets of SIP request types (known as methods) basically defined in RFC 3261[1], such as ACK, REGISTER, and OPTIONS. There are also other extensions of SIP methods defined in separate RFCs or Internet drafts such as INFO, UPDATE, PRACK, and NOTIFY [2][13]. Our proposed methods to detect the INVITE, CANCEL, and BYE flooding attacks described in Section 4 can be easily extended with very slight modifications in order to detect flooding attacks by the other sets of SIP methods.

In order to extend our proposed method to those SIP methods, each SIP method is associated with the stages before, during, and after session setup.

In the stage before session setup (SBSS), UAs can register their URIs to Registrar Server (REGISTER), query the capabilities and status of other UAs and servers, or initiate session setup (INVITE). Since the request messages in SBSS can be sent from any possible UAs as in INVITE, the flooding attacks by the methods in SBSS can be detected using the proposed Algorithm 1 with proper extensions.

The stage during session setup (SDSS) is where a session setup is in progress but not yet established. In SDSS, UA can cancel the pending session (CANCEL), provisional response acknowledgement can be carried out (PRACK), or session information can be updated (UPDATE). The messages in SDSS can be sent from UAs with pending sessions only. The flooding attacks by the messages in SDSS can be detected by extending Algorithm 3 as described in Section 4.3

The stage after session setup (SASS) is where session establishment has been completed. In SASS, the session can be terminated (BYE), the auxiliary information of the session can be carried out (INFO), or the session can be transferred to another party (REFER). The messages from the methods in SASS can be generated by UAs with established sessions. Accordingly, Algorithm 2 with proper extensions can be used to detect flooding attacks by the methods in SASS.

In **Table 1**, the extensibility of the proposed detection algorithms to various SIP methods is

summarized. New SIP methods are continuously being proposed to add additional functionality to the protocol. The newly defined SIP methods may be used for flooding attacks by malicious attackers. However, the flooding attacks can be also detected by the proper extensions of the proposed detection algorithms according to their working stages, as shown in **Table 1**.

Table 1. Extension of the proposed algorithms to detect flooding attacks by various SIP methods

Stage	SBSS (before session setup)	SDSS (during session setup)	SASS (after session setup)
SIP methods that can cause flooding attacks	INVITE OPTIONS REGISTER	ACK CANCEL PRACK UPDATE	BYE INFO REFER
Detection Algorithm	Algorithm 1	Algorithm 3	Algorithm 2

7. Conclusion

In this paper, we proposed effective methods to detect SIP flooding attacks based on the upper bound of the possible number of SIP messages, considering not only the network congestion status but also the different properties of the SIP message types. As mentioned before, while the existing schemes have been focused on INVITE flooding attacks, the proposed method can be applied to various other flooding attacks by the possible SIP methods.

It is expected that when numerous multimedia services using SIP are created and offered, the security threats from flooding attacks will increase. As we have shown, the proposed algorithms can counteract the new types of flooding attacks. Likewise, the proposed method will contribute to provision of SIP-based services and applications.

References

[1] J. Rosenberg, H. Schulzrinne, G. Cvamarillo, A. Johnston, J. Peterson, R. Spark, M. Handley, and E. Schooler, “SIP : Session Initiation Protocol,” RFC 3261, June 2002.

[2] D. Sisalem, J. Kuthan, and S. Ehlert, “Denial of Service Attacks Targeting a SIP VoIP Infrastructure: Attack Scenarios and Prevention Mechanisms,” *IEEE Network*, Vol.20, No.5, pp.26-31, Sept./Oct. 2006.

[3] D. Sisalem, J. Floroiu, J. Kuthan, U. Abend, and H. Schulzrinne, “SIP Security,” John Wiley & Sons Ltd., 2009.

[4] A. Bremler-Barr and R. Halachmi-Bekel, “Unregister attacks in SIP,” *IEEE 2nd Workshop on Secure Network Protocols 2006*, Santa Barbara, CA., Nov. 2006.

[5] F. Wang and Y. Zhang, “A New Provably Secure Authentication and Key Agreement Mechanism for SIP Using Certificateless Pubilic-Key Cryptography,” *IEEE ICCIS’2007*, Dec. 2007.

[6] D. Geneiatakis and C. Lambrinoudakis, “A lightweight protection mechanism against signaling attacks in a SIP-based VoIP environment,” *Telecommunication System*, Vol.36, No.4, pp.1018-4864, Dec. 2007.

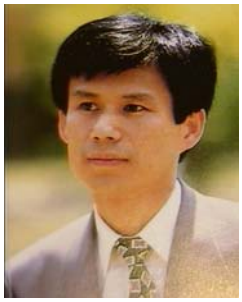
[7] Y. Rebahi, M. Sher, and T. Magedanz, “Detecting Flooding Attack against IP Multimedia Subsystem (IMS) Network,” *IEEE AICCSA’2008*, April 2008.

[8] H. Sengar, H. Wang, D. Wijesekera, and S. Jajodia, “Detecting VoIP Floods Using the Hellinger Distance,” *IEEE Trans. Parallel and Distributed Systems*, Vol. 19, No. 6, pp.794-805, June 2008.

[9] V. Siris and F. Papagalou, "Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks," *Computer Communications*, Vol. 29, No. 9, pp. 1433-1442, May 2006.
[10] M. A. Akbar, Z. Tariq, and M. Farooq, "A comparative study of anomaly detection algorithms for detection of SIP flooding in IMS," *IEEE IMSAA'2008*, Dec. 2008.
[11] Y. Ding and G. Su, "Intrusion Detection System for Signal Based SIP Attacks Through Timed HCPN," *IEEE ARES'07*, Apr. 2007.
[12] S. Kim and B. Roh, "Fast Detection of Distributed Global Scale Network Attack Symptoms and Patterns in High-speed Backbone Networks," *KSII Tr. Internet and Information Systems*, Vol.2, No.3, pp.135-149, Jun. 2008.
[13] H. Sinnreich and A. B. Johnston, "Internet Communications Using SIPs: Delivering VoIP and Multimedia Services with Session Initiation Protocol," 2nd Ed., Wiley Publishing, Inc., 2006.

Jea-Tek Ryu received his B.S. and M.S.degrees in computer science from Ajou University, Suwon, Korea, in 2005 and 2007, respectively. Since 2007, he has been a Ph.D. student at the Graduate School of Information and Communication, Ajou University, South Korea. His research interests include ubiquitous sensor networks, network security and multimedia network systems.

Byeong-hee Roh received his B.S. degree in electronics engineering from Hanyang University, Seoul, Korea, in 1987 and his M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1989 and 1998, respectively. From 1989 to 1994, he was with the Telecommunication Networks Laboratory, Korea Telecom, as a researcher. From February 1998 to March 2000, he worked with Samsung Electronics Co., Ltd., Korea, as a Senior Engineer. Since March 2000, he has been with the Graduate School of Information and Communication, Ajou University, Suwon, Korea, where he is currently an associate professor. During 2005, he was a visiting associate professor at the Dept. of Computer Science, State University of New York at Stony Brook, New York, USA. His research interests include the areas of mobile multimedia networking, network QoS, wireless sensor networks, network security and military communications.



Ki-Yeol Ryu received his B.E. degree at the Department of Computer Engineering from the Seoul National University in 1985, and his M.S. and Ph. D. degrees at the Department of Computer Science from KAIST in 1987 and 1992, respectively. He was a researcher at the Department of Computer Science, KAIST from 1992-1993. From 1993-1994, he was a visiting researcher at Yonezawa Lab., the Department of Information Science, Tokyo University. He was a visiting professor at the Department of Computer Science, University of Colorado, Boulder during 2000. He has been working at the Division of Information and Computer Engineering, Ajou University, since 1994. His interests include ubiquitous computing, service-oriented computing, component models and frameworks, object-oriented programming languages and computer security.