Detect SIP Flooding Attacks in VoLTE by Utilizing and Compressing Counting Bloom Filter

Mingli Wu[1], Na Ruan[1(✉)], Shiheng Ma[1], Haojin Zhu[1], Weijia Jia[1], Qingshui Xue[2], and Songyang Wu[3]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China
{mingliwu,ma-shh}@sjtu.edu.cn, {naruan,zhu-hj,jia-wj}@cs.sjtu.edu.cn
[2] School of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai 200240, China
xue-qsh@sit.edu.cn
[3] The Third Research Institute of Ministry of Public Security, Shanghai 200240, China
wusongyang@stars.org.cn

Abstract. As a new generation voice service, Voice over LTE (VoLTE) has attracted worldwide attentions in both the academia and industry. Different from the traditional voice call based on circuit-switched (CS), VoLTE evolves into the packet-switched (PS) field, which is quite open to the public. Though designed rigorously, similar to VoIP service, VoLTE also suffers from SIP (Session Initiation Protocal) flooding attacks. In this paper, two schemes inspired by Counting Bloom Filter (CBF) are proposed to thwart these attacks. In scheme I, we leverage CBF to accomplish flooding attack detection. In scheme II, we design a versatile CBF-like structure, PFilter, to achieve the same goal. Compared with previous relevant works, our detection schemes gain advantages in many aspects including low-rate flooding attack and stealthy flooding attack. Moreover, not only can our schemes detect the attacks with high accuracy, but also find out the attacker to ensure normal operation of VoLTE. Extensive experiments are performed to well evaluate the performance of the proposed two schemes.

Keywords: SIP flooding attack · CBF · Count · Filter

1 Introduction

As a voice call paradigm, VoLTE has attracted worldwide attentions of the public. Different from the traditional CS call, VoLTE evolves into PS field, determining to provide more reliable and rich user experience. The transition brings many benefits, such as multimedia support including high quality voice and video call, less set-up time, and less end-to-end delay. Also, compared with VoIP, which

L. Ma et al. (Eds.): WASA 2017, LNCS 10251, pp. 124–135, 2017.
DOI: 10.1007/978-3-319-60033-8_12

has dominated in PS voice telecommunication services, VoLTE gains its obvious advantages in higher voice quality, less drop-out rate, and faster set-up time for dedicated LTE resource reservation. However, the prevalence of VoLTE also involves it into various attacks, especially flooding attacks exploiting the spoofed SIP messages attempting to undermine the IMS (IP Multimedia System) or UEs (User Equipments), which is a tricky problem remaining to be solved.

Kim et al. [1] successfully exploit the SIP signal bearer in VoLTE to achieve free data transmission in forms of Mobile-to-Mobile and Mobile-to-Internet. Same loopholes are also revealed in [2]. Since the dedicated VoLTE SIP signal bearer is free and bandwidth reserved [2], even normal users would be tempted to send data through it, resulting in flooding attacks to IMS.

In case of SIP flooding attack detection, many works have been proposed. Tang et al. [3,4] propose a SIP flooding attacks detection and prevention scheme by integrating a three-dimensional sketch design with the Hellinger Distance (HD) technique. One obvious drawback of their scheme is that it needs a training period lasting even for $10\,\text{s} \times 10 = 100\,\text{s}$. However, in the case of attacks may occur at any time, it is impractical to ensure the training set is not contaminated by vicious SIP messages. Another drawback is that it is incapable to detect stealthy flooding attack. Stealthy flooding attack is a kind of attack that is difficult to be distinguished because the attacker patiently increases the flooding rate in slow pace. Sengar et al. [5,6] also propose the statistical detection mechanism called vFDS based on sudden surge caused by incomplete the handshaking processes in SIP. In their scheme, training phase is also needed to provide a baseline.

In order to thwart the above serious flooding attacks, in this paper, we propose two novel flooding attack detection schemes enlightened by CBF, a data structure widely used in many fields. In scheme I, we demonstrate there are plenty of remaining spaces in it to be exploited to detect the SIP flooding anomalies. To utilize CBF to accomplish our detection goal, we illustrate the lower bound of CBF and evaluate the overflow risk caused by repeated insertions of elements. However, due to the limited size of unit counter in CBF, there are still concerns whether the repeated insertion will lead to new overflow issue of CBF, which has been discussed in Fan's work [7]. On basis of his work [7], proof of concept is conducted to validate the low overflow risk due to repeated insertions.

To extend the detection capability concerned about multi-attributes flooding attack, we propose scheme II. Inspired by CBF, we create our own data structure named PFilter to detect the attacks. PFilter is like horizontally compressed CBF and gains strong capability in filtering SIP messages. PFilter is able to filter out large portion of normal SIP messages and prevent suspicious ones by virtue of a dynamic threshold. To get an appropriate threshold, we take exponentially weighted moving average (EWMA) to estimate the normal average transmission level during sampling period.

To sum up, the contributions we make in this paper are as follows:

(1) We demonstrate the remaining capacity of Counting Bloom Filter can be utilized to thwart SIP flooding attack.

(2) We design PFilter, a versatile structure, which gains great capability to filter out a large portion of normal SIP stream and prevent vicious messages.
(3) Extensive experiments are implemented to evaluate the performance of our two schemes, and results demonstrate their effectiveness.

The remainder of this paper is organized as follows. Section 2 introduces the preliminaries of our detection schemes. Section 3 describes the attack model and our two SIP flooding detection schemes. In Sect. 4, we perform our experiments and evaluate the performances of our schemes. Section 5 reviews the prior related works. Finally, in Sect. 6, we conclude this paper.

2 Preliminaries

In this section, Counting Bloom Filter will be introduced from its structure, parameter configuration to overflow issue.

A Bloom Filter is a space-efficient probabilistic data structure to test whether an element is a member of a particular set. The feather of Bloom Filter is that false positive (e.g., an element not in the set but wrongly being taken as a set member) matches are possible while false negative (e.g., an element in a set but not being taken as a set member) ones are not. The false positive probability of Bloom Filter is

$$f = (1 - e^{-nk/m})^k \tag{1}$$

where n is the number of elements in the set, m is the length of the bit array, and k is the number of hash functions.

Given m and n, the optimal value of k that minimizes f is

$$k = \frac{m}{n} \ln 2 \tag{2}$$

The idea of Bloom Filter is that it uses k independent hash functions h_1, h_2, $\cdots$, h_k to hash each item x_i in the set to position $h_1(x_i)$, $h_2(x_i)$, $\cdots$, $h_k(x_i)$ in a bit array of m bits that are initiated as 0. The hashed bits are set to 1 and the range of this array is $\{0, 1, 2, \cdots, m-1\}$. When examining whether an element belongs to this set, one can just check the k corresponding bits. Only all the k corresponding bits are 1 will the element be taken as a legal element, otherwise not. It does not support element deletion, Fan et al. [7] suggest Counting Bloom Filter to remedy this defect by adding a counter to record the number of each bit in the bit array. When deleting an element, the numbers of the corresponding bits of the element in the k counters will decrease by 1. The corresponding numbers will increase by 1 for add.

Practically, the arithmetic overflow due to the limited size of each counter in CBF is also an important factor supposed to be considered. In CBF, the 4-bits counter can only support 15 insertions at most during a period. Once a particular position has been hashed to more than 15 times, then the counter will overflow, resulting in adverse implications to the later operation on CBF. The probability that any count is greater or equal to i is

$$Pr(max(c) \geq i) \leq m \cdot (\frac{eln2}{i})^i \tag{3}$$

In their work, the authors demonstrate that the probability of overflow is minuscule when allowing 4-bits per counter. In this paper, we will take the unit length as 4-bits following the most common practice.

3 Attacks and Defense Mechanisms

In this section, we will describe the attack model and the two defending schemes. In scheme I, we take CBF to authenticate the incoming SIP messages and exploit its remaining capacity to thwart the flooding attacks. To remedy its poor scalability in detecting multi-attributes flooding attack, we design PFilter, a compressed CBF-like data structure we utilize in scheme II.

3.1 Attack Model

In VoLTE, the attacker is able to craft the source information of SIP messages to avoid being captured. They launch flooding attack on IMS by transmitting excessive INVITE messages at the victims' identities. In addition to INVITE, other SIP attributes, such as ACK, BYE, REGISTER, can also be exploited to mount flooding attacks. If the attacker simultaneously floods multiple SIP attributes messages, then it will result in multi-attributes flooding attack.

3.2 SIP Message Authentication

Since attackers are capable of crafting SIP messages by modifying the source information, an intuitive and effective method is to authenticate every SIP message. To achieve this authentication goal, very SIP message should carry a secret key released by the VoLTE carrier and is supposed to be updated periodically. The key agreement could be accomplished through IMS-AKA suggested in GSMA official document. The key can be the signature of each UE in VoLTE. All the secret signatures should be stored in CBF to achieve the authentication goal and the IMS server will check all signatures the coming SIP messages carry by referring CBF. Therefore, the attacker cannot be an imposter who sending SIP messages at other victims' identity.

3.3 Defense Scheme I: Simply Count by CBF

In VoLTE SIP flooding attacks, abnormal users and vicious attackers always attempt to send excessive SIP messages to achieve most benefits. In addition to the authentication function, CBF gains an advantageous ability to count the number of repeated insertions for an element during a certain period. We demonstrate this property of CBF and take advantage of its remaining capacity to detect the flooding messages.

A. Remaining Capacity

Though each counter in CBF can only be counted for at most 15 times and have already been counted for its primary counting usage, it still remains sufficient remaining spaces for us to count the incoming SIP messages.

For an arbitrary counter in CBF, the probability that it holds count i is

$$P(i) = \binom{n \cdot k}{i} \cdot (\frac{1}{m})^i \cdot (1 - \frac{1}{m})^{n \cdot k - i} \tag{4}$$

According to Eqs. (2) and (4), the expected number of counters that hold count i is

$$\begin{aligned} c_i &= m \cdot P(i) \\ &= m \cdot \binom{n \cdot k}{i} \cdot (\frac{1}{m})^i \cdot (1 - \frac{1}{m})^{n \cdot k - i} \qquad (5) \\ &= m \cdot \binom{m \ln 2}{i} \cdot (\frac{1}{m})^i \cdot (1 - \frac{1}{m})^{m \ln 2 - i} \qquad (6) \end{aligned}$$

Thus, the ratio that the counters holding i account for all the none-zero counter is

$$h(i) = \frac{c_i}{m - c_0}$$

Then the remaining capacity ratio for that counters containing i is

$$r(i) = h(15 - i) = \frac{c_{15-i}}{m - c_0}$$

Given a large $m = 33,547,705$(how this number is chosen will be explained in the evaluation part), then we can see $r(14) \approx 0.693$, $r(13) \approx 0.240$, $r(12) \approx 0.056$, $r(11) \approx 0.010$, $r(10) \approx 0.001$. Therefore, $r(i \geq 10)$ almost equals 1. It means that in CBF, counters whose remaining capacity are no less than 10 are almost 100%. However, there are still two factors supposed to take into account. One is the lower bound concern and the other is the overflow issue.

B. Lower Bound of CBF

For each element in a particular set that has been stored in CBF, the lower bound is defined as the minimum number its k hashed counters contain.

Theorem 1. *Assuming that $k \geq 12$, then among the k counters that a legal element hashed to Counting Bloom Filter, there is high probability that at least one counter contains 1, where m is the length of the counters in Counting Bloom Filter, m_0 and m_1 is the number of counters containing 0 and 1 respectively, k is the number of hash functions.*

Proof. In Counting Bloom Filter, for a legal element, the probability that at least one counter in its k hashed counters contains 1 is

$$\begin{aligned} p &= 1 - \binom{k}{0} \cdot (\frac{m_1}{m - m_0})^0 \cdot (1 - \frac{m_1}{m - m_0})^k \\ &= 1 - (1 - \frac{m_1}{m - m_0})^k \qquad (7) \end{aligned}$$

As long as $(1-\frac{m_1}{m-m_0})^k$ is small enough, then we can deduce that $p \approx 1$. Since c_i is an expected value of m_i, we can use c_0 and c_1 to estimate the m_0 and m_1. Given $m = 33,547,705$, then we know $c_0 = 1.69 \times 10^7$ and $c_1 = 1.16 \times 10^7$. Substitute m_0 and m_1 with c_0 and c_1 respectively in (7), then we can get

$$p = 1 - 0.303929^k \tag{8}$$

Since k is an integer, if we expect $p < 0.999999$, then we can easily get $k \geq 12$. It implies that if $k \geq 12$, the probability that at least one of the k counters contain 1 will approach 1.

According to Theorem 1, the probability that every legal element in the set corresponds to at least one counter contains 1 is high. This is a great property can be utilized to count how many SIP messages a UE/subscriber has sent during a sampling period, for this particular counter only belongs to this element, otherwise the original count before the repeated insertion into this counter will exceed 1.

C. Overflow Risk

Though CBF gains a good lower bound, there are still concerns whether the repeated insertions of elements will cause the overflow issue because of the limited 4-bit size per counter. Based on formula (3), we can further deduce that

$$\begin{aligned}
Pr(max(c) \geq 15) &\leq 1.0579 \times 10^{-14} \times m \\
Pr(max(c) \geq 14) &\leq 6.3957 \times 10^{-13} \times m \\
Pr(max(c) \geq 13) &\leq 1.2454 \times 10^{-11} \times m \\
Pr(max(c) \geq 12) &\leq 1.2452 \times 10^{-10} \times m \\
Pr(max(c) \geq 11) &\leq 3.7239 \times 10^{-9} \times m
\end{aligned}$$

Still, given $m = 33,547,705$, then $Pr(max(c) \geq 11) \leq 0.125$. Since the remaining capacity of CBF $r(i \geq 10) \approx 1$, then even when $Rmax = 4$, the probability that CBF will overflow is low.

Though the overflow risk is low, we take our ejection strategy to eliminate the negative effects of malicious messages. When the lower bound for a message exceeds $Rmax$, it implies this message is an abnormal one and we will delete all its previous insertions to eject it, then put it into a blacklist to prevent its further intrusion.

3.4 Defense Scheme II: Count by Lightweight PFilter

Scheme I is capable of detecting any abnormal user who has sent excessive SIP messages. However, due to the memory consumption, it cannot be easily scaled to detect multi-attributes flooding attack and still hold the overflow risk even it is low. Therefore, in this part, we propose another effective scheme based on PFilter, a data structure inspired by CBF. It is noteworthy that the length of PFilter is far less than that of CBF, so it is lightweight.

A. PFilter

PFilter is like a horizontally compressed CBF that also exploits k_p hash functions $h_1, h_2, \cdots h_{k_p}$ to profile each element x_i into position $h_1(x_i)$, $h_2(x_i)$, $\cdots$ $h_{k_p(x_i)}$ of an array with range $\{0, 1, 2, \cdots, m_p - 1\}$. When a SIP message comes, the system will extract the signature of it and profile it into PFilter. In CBF, counters still holding 0 accounter for a large part of CBF. However, PFilter does not rely on the 0 counters but a threshold to accomplish the judgement. Therefore, it is much more space efficient. The tricky question arises how to choose a good and reliable threshold, which is critical to our detection effects.

B. Filter Threshold

Since the SIP flooding attacker always intends to send excessive messages, these malicious messages will outnumber normal users and deviate from the normal level. By virtue of PFilter, we can compact all messages into it and find out the outlier. The tricky question arises how to choose a good and reliable threshold, which is critical to our detection effects. Fortunately, we find that EWMA is pretty appropriate to create the dynamic threshold adapted with the stochastic SIP stream.

Denote α_i as the measured average number of messages each VoLTE user sends during sample round i, R_i as the estimated one and β_i as the average skewed distance between the α_i and R_i. Then

$$R_i = (1 - \lambda_1) \cdot R_{i-1} + \lambda_1 \cdot \alpha_i \tag{9}$$

Because the traffic is frequently fluctuate over time, we are also supposed to estimate the skewed distance

$$\beta_i = (1 - \lambda_2) \cdot \beta_{i-1} + \lambda_2 \cdot |\alpha_i - R_i| \tag{10}$$

In (9), the average measured transmission times is

$$\alpha_i = \frac{N_i}{U_i}$$

where N_i is the message number, U_i is the caller number in round $i (i \geq 1)$. And $0 < \lambda_1 \leq 1$, $0 < \lambda_2 \leq 1$ are the weight factors. In (9) and (10), λ_1 and λ_2 are constant factors that determine the memory depth of EWMA. The closer they are to 1, the more weight EWMA lays in the current measurement. A value of $\lambda_1 = 1$(or $\lambda_2 = 1$) implies EWMA only cares about the current measurement.

Given the estimated average threshold R_i and the estimated average skewed distance β_i, we can further calculate the average number of messages each counter in PFilter holds during sampling period i is

$$Thre_i = \frac{k_p \cdot U_i}{m_p} \cdot min\{R_{i-1} + \lambda_3 \cdot \beta_{i-1}, Rmax_i\} \tag{11}$$

$\lambda_3 \geq 1$ is a magnification factor of skewed distance and $Rmax_i$ is the maximum number of messages a legal UE can transfer. Note that since $Thre_i$ is the threshold, $Thre_i < 1$ is not allowed, otherwise it will be automatically revised to 1 in case of threshold disfunction caused by low SIP stream.

To prevent threshold pollution, we take a self-adapted strategy that $Thre$ will only be updated according to formula (9) (10) on condition that there is no attack being detected during the current sampling period, otherwise it will keep its own value.

C. Messages Filter

PFilter takes its responsibility to filter out flooding messages from the normal ones. In analogy with CBF, we take the similar strategy that as long as one of k_p counters contains a count less than the threshold $Thre$, then this message will be taken as a normal message. The reason why we can take this strategy is flooding SIP messages will conspicuously stand out in normal messages crowds. The observation is that the more drastic the flooding attack goes, the more prominent the attack messages become compared with normal messages. Even for low rate flooding attack, they will still outnumber the normal ones, thus crossing the line of PFilter.

4 Experiments and Evaluation

4.1 Experiment Set up

To evaluate our proposed mechanisms, we design our testbed comprised of three computers. In this design, one computer plays the role of normal users by sending normal SIP messages, another one as IMS server in VoLTE handling the incoming SIP messages. The third computer functions as attacker sending flooding SIP messages. We perform our defense mechanisms on the computer playing as IMS server.

4.2 Evaluation

In our scheme, we empirically choose $\lambda_1 = \lambda_2 = 0.8$. For $Thre$, we set $\lambda_3 = 2$ to slightly enlarge the skew distance and we set the maximum transmission times as $Rmax = 4$. For hash functions, we take *MurmurHash3* functions with independent seeds. *MurmurHash3* function is non-cryptographic hash function used for hash-based lookups. It has been widely deployed in many famous applications, such as Hadoop, libstdc++, Nginx. One more benefit of *MurmurHash3* is that it cares nothing about the length of input.

We randomly mount INVITE flooding attacks with varying flooding rate from 10 cps(call per second) to 100 cps. For each flooding rate, we perform 500 attacks to obtain a good evaluation of PFilter. It is noteworthy that the normal call generation rate randomly varies from 700 cps to 3200 cps, which is much more frequent than most relevant works. We take the extreme values to thoroughly evaluate the performance of PFilter.

A. Scheme I: Simply Count by CBF

In scheme I, we randomly generate $n = 1,000,000$ UEs in VoLTE and each UE maintains its unique secret signature. We also set the tolerated false positive

rate as $f = 10^{-7}$. Then according to formula (1) and (2), we can get the length of CBF is $m = 3,354,7705$ (it is the reason we take this number in Sect. 3.1) and the number of hash functions is $k = 23$. According to formula (8), it is obvious that $k = 23 > 12$ and the lower bound for CBF that the probability at least one of the k counters for a legal message contains 1 indeed approaches $p = 1$. We also compare real measured $\frac{m_i}{m}$ with $\frac{c_i}{m}$ to examine the validity of our estimation to substitute m_i with c_i in (8). It turns out the two sets of data share great similarity, and the results are shown in Table 1. The similarity also confirms the selected hash functions perform in good state.

Table 1. A comparison between $c_i/m(\%)$ and $m_i/m(\%)$

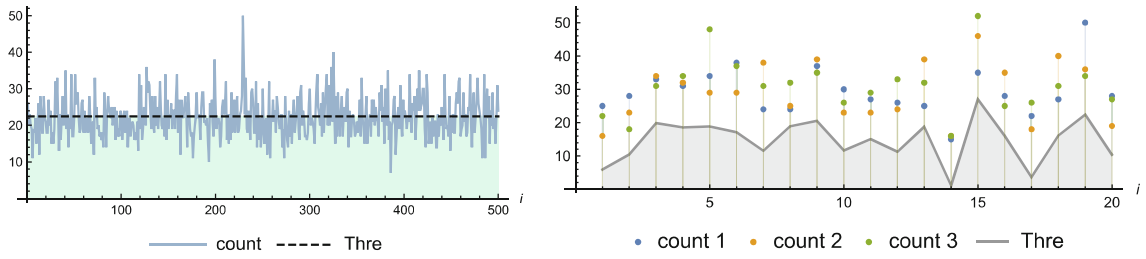
i	1	2	3	4	5	6	7	8	9	10
c_i/m	34.540	11.840	2.706	0.464	0.064	0.007	$<10^{-3}$	$<10^{-4}$	$<10^{-5}$	$<10^{-6}$
m_i/m	34.546	11.835	2.705	0.465	0.063	0.007	$<10^{-3}$	$<10^{-4}$	$<10^{-5}$	0

As long as the lower bound of one message exceeds $Rmax$, then this message will be taken as illegal one. In our more than five thousands flooding attacks at varying rate from 10 cps to 100 cps, the detection rates are always 100%, the false detection rates stay 0, and no overflow issue resulting from the repeated insertion is detected. It is not a surprise because the functional CBF relies on an absolute counting strategy and we also exploit the timely ejection approach to eradicate the adverse effects incurred by the flooding messages. Therefore, for one attribute flooding attack in SIP, CBF is sufficient to detect and find out the attacker. However, the cost for detecting multi-attributes flooding attack is high. For example, to detect INVITE and BYE flooding attack, an extra CBF for BYE messages should be created and the cost is $m * 4 = 134,190,820 bits$.

B. Scheme II: Count by Lightweight PFilter

In this scheme, we set $m_p = 500$ and $k_p = 3$. For each counter, we set the length as 8-bits. PFilter is able to filter out a large portion of normal SIP messages and we can easily find out the attacker from the small portion. We achieve high detection rate in different flooding rates, as is shown in Table 2. As we can see in this table, even when the flooding rate is as low as 10 cps, our scheme can efficiently detect this anomaly with detection rate (DR) at 76.4%. Figure 1 shows how PFilter functions to detect INVITE flooding attack at 15 cps. In Fig. 1(a), we get $Thre = 22.4$ by the estimation of equation (11). As long as one of the three entries contains a count less than 22.4, the message will be filtered out as a normal one. The dynamic feather of $Thre$ is depicted in Fig. 1(b) for 20 flooding attacks, with all the $k_p = 3$ counts of the attack message exceeding $Thre$. Because of this feather, $Thre$ can be self-adapted with the high random and fluctuant SIP traffic.

We also compare our detection results with Tang's work [3] even when we get the measurements in extreme condition describing in the above section.



(a) Utilize *Thre* to filter messages in a round

(b) *Thre* varies between rounds

Fig. 1. An example of how *Thre* works for PFilter

The results can be found in Table 2. Our scheme could still detect flooding rate at 15 cps with DR at 97.1% even when normal VoLTE call randomly fluctuates between 700 cps and 3200 cps, compared with 88% in Tang's between 30 cps and 80 cps. The reason why we do not choose more drastic flooding rate as [3] is that since our scheme can detect flooding rate at 100 cps, it is certain we can detect more drastic flooding attack.

Table 2. Detection results: Tang's [3] *vs* Our's

Flooding rate	DR (Tang's)	DR (Our's)
10	-	76.4%
15	88%	97.1%
35	100%	100%
50	100%	100%
75	100%	100%
100	100%	100%
500	100%	-

Compared with scheme I, the low-rate flooding attack detection rate in scheme II is lower. However, when it comes to the multi-attributes flooding attack detection, we can simply apply other PFilters to detect other SIP attributes anomalies at small costs because it is much more memory saving compared with CBF. For example, in addition to INVITE flooding attack detection that costs $m_p \times 8 = 4000bits$, if the attacker also launch BYE flooding attack, the memory cost will be another $4000bits$. It is obvious that $4000 \times 2bits < 134,190,820bits$, so it is much cheaper than scheme I.

5 Related Work

Generally, network-based intrusion detection systems can be divided into two categories: signature-based NIDSs and anomaly-based NIDSs [8].

Signature-based NIDSs rely on a context-aware "blacklist" containing various signatures that describe known attacks. Many signature-based NIDSs adopt Bloom Filter to solve the storage and computation issues. Roh et al. [9] propose whitelist-based countermeasure scheme based on none-member ratio by utilizing CBF. Geneiatakis et al. [10,11] take advantage of CBF to calculate session distance of SIP to detect anomalies with the assumption that flooding attack is associated with incomplete sessions.

Tang and Cheng [12] address the stealthy attack by combining sketch with wavelet techniques. Akbar et al. [13] leverage Hellinger distance to low rate and multi-attributes DDoS attack. In Golait and Hubballi's work [14], the authors also detect the anomaly by generating the normal profile of SIP messages as a probability distribution.

Ryu et al. [15] derive the upper bound of the possible number of SIP messages, and detect the SIP flooding attacks by checking whether this upper bound has been challenged. Mehić et al. [16] also calculate the maximum number and type of SIP messages that can be transferred during established VoIP call without raising an alarm from IDS (Intrusion detection system).

6 Conclusion

In this paper, we propose two effective schemes to detect and prevent SIP flooding attack. In scheme I, we demonstrate CBF can be exploited to keep track of how many SIP messages a VoLTE user transmit during a certain period, thus detecting the attacker. In scheme II, we design PFilter, a more lightweight data structure, to accomplish the detection goal. Not only can our schemes detect the flooding anomalies, but also find out the attackers to alleviate their adverse effects. Another advantage is that no training period is needed, so both schemes have no fear of baseline pollution. Also, the two schemes function well in low-rate flooding attack detection and keep immune to stealthy flooding attack.

Acknowledgments. This work is supported by Chinese National Research Fund (NSFC) Key Project No. 61532013; National China 973 Project No. 2015CB352401; Shanghai Scientific Innovation Act of STCSM No.15JC1402400; 985 Project of Shanghai Jiao Tong University with No. WF220103001; SIT Collaborative innovation platform under Grant No. 3921NH166033; NSFC No. 61170227 and No. 61672350.

References

1. Kim, H., Kim, D.,. Kwon, M., Han, H., Jang, Y., Han, D., Kim, T., Kim, Y.: Breaking and fixing VoLTE: exploiting hidden data channels and mis-implementations. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 328–339. ACM (2015)
2. Li, C.-Y., Tu, G.-H., Peng, C., Yuan, Z., Li, Y., Lu, S., Wang, X.: Insecurity of voice solution volte in LTE mobile networks. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 316–327. ACM (2015)

3. Tang, J., Cheng, Y., Hao, Y.: Detection and prevention of SIP flooding attacks in voice over IP networks. In: 2012 Proceedings IEEE INFOCOM, pp. 1161–1169. IEEE (2012)
4. Tang, J., Cheng, Y., Hao, Y., Song, W.: SIP flooding attack detection with a multi-dimensional sketch design. IEEE Trans. Depend. Secur. Comput. **11**(6), 582–595 (2014)
5. Sengar, H., Wang, H., Wijesekera, D., Jajodia, S.: Fast detection of denial-of-service attacks on ip telephony. In: 2006 14th IEEE International Workshop on Quality of Service, pp. 199–208. IEEE (2006)
6. Sengar, H., Wang, H., Wijesekera, D., Jajodia, S.: Detecting VoIP floods using the hellinger distance. IEEE Trans. Parallel Distrib. Syst. **19**(6), 794–805 (2008)
7. Fan, L., Cao, P., Almeida, J., Broder, A.Z.: Summary cache: a scalable wide-area web cache sharing protocol. IEEE/ACM Trans. Netw. (TON) **8**(3), 281–293 (2000)
8. Meng, W., Li, W., Kwok, L.-F.: EFM: enhancing the performance of signature-based network intrusion detection systems using enhanced filter mechanism. Comput. Secur. **43**, 189–204 (2014)
9. Roh, B., Kim, J.W., Ryu, K.-Y., Ryu, J.-T.: A whitelist-based countermeasure scheme using a bloom filter against SIP flooding attacks. Comput. Secur. **37**, 46–61 (2013)
10. Geneiatakis, D., Vrakas, N., Lambrinoudakis, C.: Performance evaluation of a flooding detection mechanism for VoIP networks. In: 2009 16th International Conference on Systems, Signals and Image Processing, pp. 1–5. IEEE (2009)
11. Geneiatakis, D., Vrakas, N., Lambrinoudakis, C.: Utilizing bloom filters for detecting flooding attacks against SIP based services. Comput. Secur. **28**(7), 578–591 (2009)
12. Tang, J., Cheng, Y.: Quick detection of stealthy SIP flooding attacks in VoIP networks. In: 2011 IEEE International Conference on Communications (ICC), pp. 1–5. IEEE (2011)
13. Akbar, A., Basha, S.M., Sattar, S.A.: Leveraging the SIP load balancer to detect and mitigate DDos attacks. In: 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), pp. 1204–1208. IEEE (2015)
14. Golait, D., Hubballi, N.: VoIPFD: voice over IP flooding detection. In: 2016 Twenty Second National Conference on Communication (NCC), pp. 1–6. IEEE (2016)
15. Ryu, J.-T., Roh, B.-H., Ryu, K.-Y.: Detection of SIP flooding attacks based on the upper bound of the possible number of SIP messages. KSII Trans. Internet Inf. Syst. **3**(5), 507–526 (2009)
16. Mehić, M., Mikulec, M., Voznak, M., Kapicak, L.: Creating covert channel using SIP. In: Dziech, A., Czyżewski, A. (eds.) International Conference on Multimedia Communications, Services and Security, pp. 182–192. Springer, Cham (2014). doi:10.1007/978-3-319-07569-3