PAPER Special Section on Security, Privacy and Anonymity in Computation, Communication and Storage Systems

Detect and Prevent SIP Flooding Attacks in VoLTE by Utilizing a Two-Tier PFilter Design

Na RUAN[†a)], ***Member***, **Mingli WU**[†], **Shiheng MA**[†], **Haojin ZHU**[†], **Weijia JIA**[†], ***and*** **Songyang WU**[††], ***Nonmembers***

SUMMARY 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 has long been open to the public. Though designed rigorously, similar to VoIP services, VoLTE also suffers from SIP (Session Initiation Protocal) flooding attacks. Due to the high performance requirement, the SIP flooding attacks in VoLTE is more difficult to defend than that in traditional VoIP service. In this paper, enlightened by Counting Bloom Filter (CBF), we design a versatile CBF-like structure, PFilter, to detect the flooding anomalies. Compared with previous relevant works, our scheme gains advantages in many aspects including detection of low-rate flooding attack and stealthy flooding attack. Moreover, not only can our scheme detect the attacks with high accuracy, but also find out the attackers to ensure normal operation of VoLTE by eliminating their negative effects. Extensive experiments are performed to well evaluate the performance of the proposed scheme.

key words: *SIP flooding attack, PFilter, 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 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 the 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. We term this kind of attack as *single-source attack* for these individual users just separately cause flooding attack. As for those advanced attackers, not only could they flood SIP messages as these abnormal users, but also maliciously launch large scale flooding attacks by simultaneously mobilizing multiple UEs in VoLTE. We term this attack as *multi-source attack*. For both misconduct users and vicious attackers, they intend to send illegitimate SIP messages as fast as they can to achieve most benefits, further aggregating the flooding attack.

Although SIP flooding attack is not new and has been known in the long suffered VoIP, it does bring new challenges to the detection scheme in VoLTE. Firstly, compared with VoIP operated by individual over-the-top (OTT) companies, VoLTE is implemented by carriers at a national level to replace the traditional CS call. To carry out VoLTE, carriers have to deploy lots of new telecommunication infrastructures. Also, smartphones need to be updated both in hardware and software to support this new telephone service. Therefore, once being attacked, the whole VoLTE system from UE to IMS will pay much more than VoIP. This puts more pressure on the detection scheme to achieve high effectiveness. Secondly, VoLTE lays more emphasis on the performance. Once been attacked by SIP flooding attack, the expected high performance will be degraded soon or even cause harm to normal UEs because of the highest priority of SIP messages [3]. Therefore, detection schemes are supposed to be efficient enough to promptly detect the anomalies to alleviate the consequences.

In case of SIP flooding attack detection, many works have been proposed. Tang et al. [4], [5] 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 in their scheme is that it needs a training period lasting even for 100*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 distinguish because the attacker patiently increases the flooding rate in slow pace. Sengar et al. [6], [7] 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. Kumar and Tilagam [8] discuss low-rate SIP flooding attack only in single source, thus not applicable to multi-

Manuscript received January 18, 2017.
Manuscript revised May 23, 2017.
Manuscript publicized July 21, 2017.

[†]The authors are with Shanghai Jiao Tong University, Shanghai, 200240 China.
[††]The author is with The Third Research Institute of Ministry of Public Security, Shanghai 201204, China.
a) E-mail: naruan@cs.sjtu.edu.cn

DOI: 10.1587/transinf.2016INP0023

source attack [9]. Tang et al. [4], [5] also emphasize that low-rate flooding attack can hardly be distinguished from normal rate fluctuation due to randomness. Because users can make VoLTE call and drop out the call at any time, the SIP stream is stochastic.

In order to thwart the above serious flooding attacks, we propose a novel flooding attack detection scheme. Inspired by CBF, we design our own data structure named PFilter to detect the attacks. PFilter gains strong capability in filtering SIP messages. In the scheme, we propose a two-tier PFilter design to achieve the detection goal. In tier 1, the PFilter shoulders the responsibility to filter out a large portion of normal SIP messages and prevent suspicious ones by virtue of the dynamic threshold. To get an appropriate threshold in this tier, we take exponentially weighted moving average (EWMA) to estimate the normal average transmission level during a period. In tier 2, another PFilter plays its role to find out the attackers. We take the message collapse algorithm to preclude the normal messages and leave the malicious ones. The threshold in this tier is configured as a static one to handle complicated attack models. Moreover, our scheme also can detect low-rate flooding attack with high accuracy, and keep immune to stealthy flooding attack at the same time. We achieve good detection results even when the flooding rate is as low as 10 cps (call per second).

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

1) We design PFilter, a versatile structure, which gains great capability to filter out a large portion of normal SIP stream and prevent vicious messages.

2) Then we propose a two-tier PFilter structure that could cooperate with each other intimately to detect the anomalies and find out the attackers.

3) Extensive experiments are implemented to evaluate the performance of our scheme, and corresponding parameters configuration is suggested to provide practical guidance to VoLTE carries.

The remainder of this paper is organized as follows. Section 2 introduces the preliminaries of our detection scheme. Section 3 presents the attack model. In Sect. 4, we describe our SIP flooding detection scheme. In Sect. 5, we perform our experiments and evaluate the performances of our scheme. Section 6 makes a comprehensive analyses of the proposed scheme. Section 7 reviews the prior related works. Finally, in Sect. 8, we conclude this paper.

2. Preliminaries

2.1 VoLTE Call Flow

A basic VoLTE call flow is as depicted in Fig. 1. First, UE_A initiates a VoLTE call by sending an INVITE message to P-CSCF (Proxy-Call Session Control Function) in his home network. Then P-CSCF forwards this message to S-CSCF (Serving-Call Session Control Function) and respond to UE_A with 100 TRYING. S-CSCF will extract the

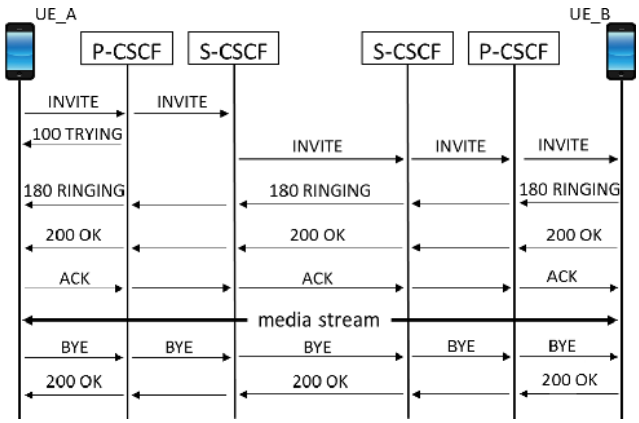


Fig. 1 A basic VoLTE call flow between UE_A and UE_B

URI information from the INVITE and send this message to home network of UE_B. In UE_B's home network, S-CSCF then send the INVITE message to the P-CSCF UE_B corresponds to. The P-CSCF will transmit this message to UE_B. After receiving the INVITE message, UE_B should respond to it by sending back a 180 RINGING message to UE_A to remind both caller and callee. Upon answering the call, UE_B will send a 200 OK to UE_A. Then UE_A will respond an ACK to UE_B. After the handshaking, the media bear will be established for voice and video stream. After finishing the call, UE_A (or UE_B) will close this session by sending a BYE message to UE_B (or UE_A). Then UE_B (or UE_A) will respond with 200 OK to release this session.

2.2 Counting Bloom Filter

A Bloom Filter is a space-efficient probabilistic data structure to test whether an element is a member of a particular set. 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 an legal element, otherwise not. It does not support element deletion, Fan et al. [10] suggest Counting Bloom Filter (CBF) to remedy this defect by adding a counter for each bit in the bit array to record how many times it has been hashed to. 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 when adding an element.

3. Attack Model

In addition to the aforementioned attacks, in this section, we introduce three kinds of common SIP flooding attack according to the SIP attributes the attackers exploit.

3.1 INVITE Flooding Attack

Since INVITE message is designed to activate a VoLTE call session, the adversaries would take advantage of this SIP attribute to launch DoS attacks. The attackers just arbitrarily flood thousands of INVITE messages to overwhelm the IMS (IP Multimedia Subsystem) or even the UEs. As we can see from Fig. 1, once triggered by INVITE requests, all the VoLTE components, such as P-CSCF, S-CSCF, will be activated. Also, by modifying source information of INVITE message and initiating VoLTE call with some VoLTE users' identities, these innocent users could be framed without any awareness.

3.2 BYE Flooding Attack

As BYE message is used to terminate the VoLTE call session. Attackers could hurl junk BYE messages to consume the resources of the VoLTE the way as INVITE messages. More severely, attackers could launch flooding attacks to specific VoLTE users to shutdown the ongoing VoLTE calls. Therefore, if this attack is mounted in a large scale, the drop-out rate in VoLTE network would surge.

3.3 Multi-Attributes Flooding Attack

In addition to the two above attacks, there is also ACK flooding attacks by bombarding the IMS in explosive mode. In the multi-attributes attack, the intelligent attackers could deluge the IMS by flooding combinations of the above SIP messages, which could result in more severe consequences. For some SIP flooding attack detection schemes relying on the correlation between different SIP attributes, this intelligent attack could frustrate them by observing the regular correlation pattern.

4. Defense Mechanisms

Since the SIP flooding attacker always intends to send excessive messages, these malicious messages will outnumber normal users and deviate from the normal level. In this section, we propose an effective scheme based on PFilter to find out the attackers. This scheme feathers in a two-tier PFilter detection structure, and each tier plays its unique role and can also efficiently cooperate with each other.

4.1 PFilter

PFilter is a CBF-like data structure that exploit 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 SIP address 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 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.

4.2 Tier 1: Suspects Filter

In this tier, we find that EWMA is appropriate to create the dynamic threshold adapted with the stochastic SIP stream.

Filter Threshold. As mentioned before, the SIP stream is highly fluctuant because users could initiate and hang on their VoLTE calls at any time. The random feather makes the SIP stream drastically fluctuate with time, thus increasing the difficulty to read its regular pattern. By virtue of PFilter, we can compact all messages into it and find out the outliers who intend to transmit excessive messages. 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 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

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

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

$$\beta_i = (1 - \lambda_2) \cdot \beta_{i-1} + \lambda_2 \cdot |\alpha_i - R_i| \tag{2}$$

In formula (1), 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 (1) and (2), λ_1 and λ_2 are constant factors that determine the memory depth of EWMA. The more close 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

$$Thre1_i = \frac{k_p \cdot U_i}{m_p} \cdot min\{R_{i-1} + \lambda_3 \cdot \beta_{i-1}, Rmax_i\} \tag{3}$$

$\lambda_3 \geq 1$ is a magnification factor of skewed distance and $Rmax_i$ is the maximum number of messages a legal UE can transfer during the sampling period. Note that since $Thre1_i$ is the threshold, $Thre1_i < 1$ 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 $Thre1$ will only be updated according to formula (1) (2) on condition that there is no attack being detected during the current sampling period, otherwise it will

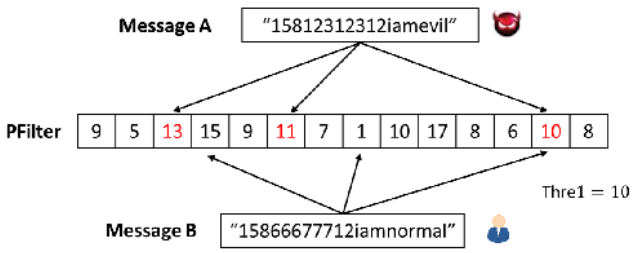


Fig. 2 An example of how PFilter distinguishes a vicious SIP message from normal ones

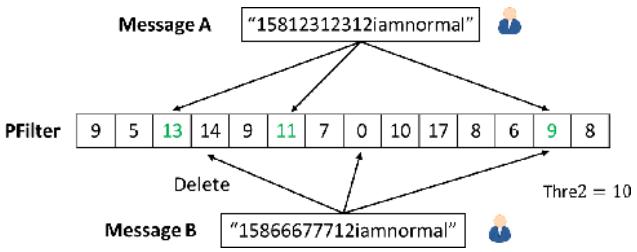


Fig. 3 Solve the misjudgement in tier 1 by deleting legal messages in advance

keep its own value. Therefore, we can train a healthy threshold to keep an effective detection state.

Messages Filter. In tier 1, 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 $Thre1$, then this message will be taken as a normal message. Figure 2 shows how PFilter performs to distinguish the vicious messages from the normal ones. 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. In Fig. 2, assuming that we choose $k_p = 3$and the threshold $Thre1$ as 10, then all three counters message A hashed to hold the counts (i.e., 13, 11, 10 respectively) all exceeding (included) the threshold, it implies that message A is abnormal. In contrast, though message B crosses the line twice (i.e., 15 and 10), there is still one corresponding counter loads a number (i.e., 1) less than the threshold, then it will be taken as a normal one.

Similar to $Thre1$, our detection scheme also keeps its memory by maintaining a blacklist. First, if a message in SIP traffic is not on the blacklist, then we profile it into PFilter according to its signature, otherwise simply drop it. Next, we check the k_p hashed counters for each message as depicted in Fig. 2. Finally, we obtain the suspicious SIP messages and goes the second tier 2 round, which shoulders the responsibility to find out the attackers.

4.3 Tier 2: Attackers Finder

Tier 1 gains the ability to filter out a large portion of messages and retain a small portion suspicious messages S. However, taking consideration of hash collision, some benign messages are still in the suspicious list, resulting in the malicious ones hiding in the SIP message crowds. In this tier, we determine to distinguish the hidden attackers from the normal ones. An intuitive method is to filter the suspicious messages S again in the same way as that in tier 1, thus further narrowing the searching scope. The defect is that it is possible the attack messages are still mixed with the normal ones. Therefore, to accelerate the attackers hunting process, we take a different filter approach based on another PFilter. Since there is only small portion of messages, we utilize our *collapse* strategy to achieve the attacker elimination goal. The key idea of this strategy is to delete the normal massages once it is taken as a legitimate message. Therefore, the attack messages are left.

In Fig. 2, it is also possible message A is not an attack message but wrongly being taken as a malicious one because of hash collision, and that is why it still remains suspicious SIP stream after the filtering process in tier 1. Therefore, to alleviate the above adverse effects, we take another strategy by repeatedly deleting a highly possible legal message from PFilter. Message B only occurs once and could be taken as a normal one with confidence. After deleting it, message A will resume to a normal state, as is illustrated in Fig. 3.

Collapse Threshold: For there is only a small portion of SIP messages, in this tier, we take a simple approach by choosing a static threshold as $Thre2 = 10$. It implies that as long as a message corresponds to one counter less than 10, it will be regarded as a normal message. We do not choose the threshold the way as $Thre1$ in tier 1. An important consideration is that it is possible all SIP messages IMS server receives during our sampling period are crafted by attackers. For example, in the middle night when seldom users make VoLTE phone calls, then PFilter in tier 1 will be ineffective to notice the anomaly by utilizing the dynamic threshold. By utilizing a static threshold, such full attackers trick can be eliminated. The other reason is that we are confident our tier 1 PFilter is capable to filter out a large portion of SIP traffic and leaves only a small part, so a static threshold is sufficient to pick out the attackers with little losses in accuracy. If we apply our collapse strategy in tier 1, the costs can be huge for repeatedly deleting elements in large mounts of messages. Therefore, by integrating the dynamic threshold in tier 1 and a static one in tier 2, our detection scheme becomes more robust.

Messages Collapse: For Pfilter in tier 2, the core idea is to delete normal messages in the suspicious messages S remaining by tier 1. The reason why we call it *collapse* is that the deletion operations will consistently reduce the height of entries in PFilter, a similar operation in the classical game *Tetris*. The key idea is to repeatedly scan the PFilter and delete the normal messages from it, thus making the attackers incapable to hide among the crowd.

5. Experiments and Evaluation

5.1 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 handling the incoming SIP messages. The third computer functions as attackers sending flooding SIP messages. We perform our defense mechanisms on the computer playing as IMS server.

5.2 Evaluation

In our scheme, we empirically choose $\lambda_1 = \lambda_2 = 0.8$. For $Thre1$, 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. *MurmurHash* 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. And the current version is *MurmurHash3*. One more benefit of *MurmurHash3* is that it cares nothing about the length of input.

We randomly mount the flooding attacks with varying flooding rate from 10 cps 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 and provide suggestions for parameter configuration.

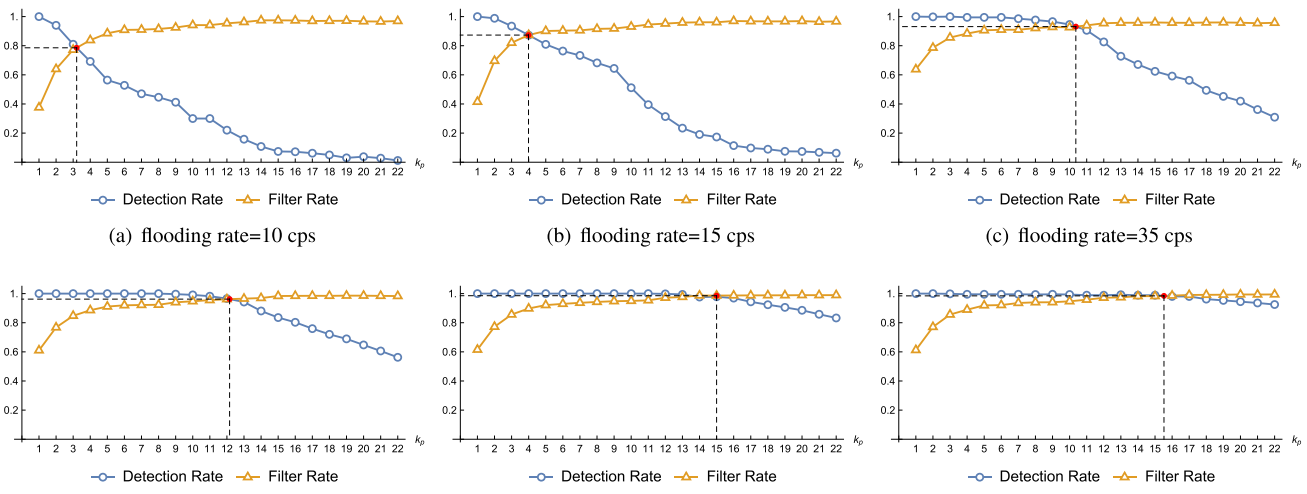
5.2.1 Number of Hash Functions for PFilter

The results is depicted in Fig. 4. In this figure, we compare the Detection Rate (DR) and Filter Rate (FR), two critical but contradictory factors for PFilter. Generally, with the increase of k_p, the Detection Rate decreases, while the Filter Rate increases. The reason is that more hash functions will make it more difficult for all the k_p corresponding counters to meet the filter threshold. However, with a larger k_p, the probability for a vicious message to not be detected also increases, especially for low-rate flooding messages who are akin to the normal ones. Ideally, a 100% DR and a approximate 100% FR for flooding detection are desired. However, it is not always true, particularly for low flooding rate, as is shown in the six subfigures. For flooding rate 10 cps and 15 cps in Fig. 4 (a) and 4 (b), DR and FR intersect at $(3.21, 78.6\%)$ and $(3.99, 87.3\%)$ respectively. Though it has already achieved a good balance, it is still less than 100%. As for flooding rate more than 75 cps, 4 (e) and 4 (f) show DR and FR can nearly converge at 100%. It implies that even by virtue of only one Pfilter in tier 1, it can accurately detect and find out the vicious messages. This property is significant for SIP flooding detection, as attackers always intend to launch more drastic flooding attacks to bombard the IMS server and UEs.

5.2.2 The Length of PFilter

The length of PFilter is also an important factor for flooding attack detection. Figure 5 shows DR and FR when m_p varies under different flooding rate. As illustrated in this figure, as m_p increases, the detection rate increases under flooding rate at 15 cps and 35 cps. When flooding rate is 50 cps, because the attack message has already been distinguished when $m_p = 100$, DR does not show a obvious increase. As for FR, all three subfigures in 5 show that m_p does not obviously affect the filter rate of PFilter.

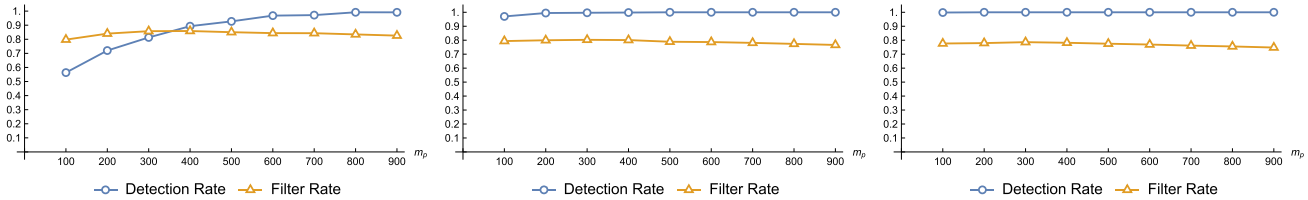
We also compare our detection results with Tang's work [4] even when we get the measurements in extreme condition describing in the above section. The results can



(a) flooding rate=10 cps (b) flooding rate=15 cps (c) flooding rate=35 cps

(d) flooding rate=50 cps (e) flooding rate=75 cps (f) flooding rate=100 cps

Fig. 4 The Filter Rate vs. Detection Rate for PFilter in tier 1 with varying k_p



(a) flooding rate=15 cps (b) flooding rate=35 cps (c) flooding rate=50 cps

Fig. 5 The Filter Rate vs. Detection Rate for PFilter in tier 1 with varying m_p

Table 1 Detection results: Tang's [4] 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%	-

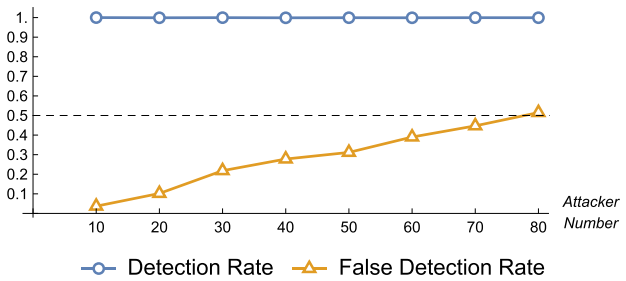


Fig. 6 Multi-source flooding attack detection rate and false detection rate

be found in Table 1. In the experiment, we choose $k_p = 3$ and $m_p = 500$ to achieve a good balance of detection rate, filter rate and memory consumption. Our scheme could still detect flooding rate at 15 cps with 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 [4] is that since our scheme can detect flooding rate at 100 cps, it is certain we can detect more drastic flooding attack.

For *multi-source attack*, our scheme is also effective to detect the attack with high accuracy. We depict the detection rate and false detection rate in Fig. 6. The flooding rate is set as 15 cps. As we can see in the figure, the detection rate always keeps close to 100% as the number of attackers increase, while the false detection rate also increases. When the attacker number increases as 80, the detection rate climbs to slightly over 50%. It means that to detect multi-source attack with 80 attackers, 80 normal VoLTE users will be erroneously judged as attackers. The reason is that more attackers will pollute more counters in PFilter with solid length, resulting in false detection. When there are only 10 attackers, the false detection rate becomes as low as 3.7%. It implies to detect 10 attackers, far less than one normal user will be misjudged.

5.3 Multi-Attributes Flooding Attack Detection

For advanced attackers who launch multi-attributes flooding attack (INVITE, 200 OK, ACK, BYE), we can simply apply our two-tier PFilter structure to detect other SIP attributes anomalies in the same way. All that need are multiple PFilters corresponding to each SIP attribute.

6. System Analysis

Effectiveness. As demonstrated in above section, our scheme could accurately detect the attacks even when flooding rate is as low as 10 cps. Also, for *single-source attack*, our scheme can find out the attacker with great confidence. It implies that abnormal users cannot utilize the VoLTE signal bear for free any longer. When *multi-source attack* occurs, the scheme will degrade as the number of attackers increases because of entries pollution. This is the main limitation and can be remedied by broadening its length. Though normal users may be mistakenly regarded as attackers due to hash collision in this condition, few attackers can escape from our capture.

Another benefit is that our detection scheme do not rely on training period, thus having no fear for malicious messages contamination. For innocent UEs being targeted by attackers, our scheme can also identify the anomalies by profiling the callee number of SIP messages in sampling windows.

Efficiency. In our scheme, 8-bits counter is enough for a PFilter to take its detection. For flooding rate more than 255 cps, the corresponding counters just freeze on this maximum state to avoid counters overflow. With $m_p = 500$, two PFilters will only cost $500 \times 8 \times 2 \times 4 = 32000$ bits for 4 attributes SIP flooding attack detection. Compared with the memory consumption in [5], which uses 40960 bits, our scheme turns cheaper. For computational cost, we perform our scheme by a computer with memory of 7.7GB and i7-4770 CPU with 3.4GHz. And the average CPU time is 34678 *ms* in case of 80 attackers. It demonstrates the overhead by implementing our scheme is low.

Scalability. The proposed scheme achieves good scalability in multi-attributes SIP attacks. For more drastic SIP flooding attack detection, PFilter can either horizontally widen its length to promote its accuracy or vertically add more tiers. What is more, we hold that our scheme could also be scaled to detect flooding attacks in other fields, such

as the RTP flooding attack.

Stealthy Attack Proof. Since our detection scheme is based on the assumption that malicious attackers always attempt to send excessive SIP messages, the attackers will be promptly detected with the assistance of a combination of dynamic and static threshold in our two-tier design once they attempt to break the cordon. Even they mount the stealthy attack with great patience, they cannot escape from the capture.

7. Related Work

Generally, network-based intrusion detection systems can be divided into two categories: signature-based NIDSs and anomaly-based NIDSs [11]. Many signature-based NIDSs adopt Bloom Filter to solve the storage and computation issues. Roh et al. [12] propose whitelist-based countermeasure scheme based on none-member ratio by utilizing CBF. Geneiatakis et al. [13], [14] take advantage of CBF to calculate session distance of SIP to detect anomalies with the assumption that flooding attack is associated with incomplete sessions and there exists correlations between different SIP attributes. Rebahi et al. [15] also consider the half-open connection issue, and propose a non-parametric CUSUM algorithm to detect gradual change in means of time series.

To prevent identity proof flooding attack, authentication is an effective way as is suggested in [1]. IPSec seems a plausible method to authenticate the packets. However, the drawback is that it needs both the communication parties's support and the overhead caused by IPSec is also a noticeable issue. It is ineffective at mitigating a bandwidth-based DoS attack, which may in fact exacerbate the problem, since initializing a connection may require more resource when using IPSec than not [16].

Tang et al. [17] address the stealthy attack by combining sketch with wavelet techniques. Akbar et al. [18] also leverage Helliger Distance to low rate and multi-attributes DDoS attack. In Golait and Hubballi's work [19], the authors also detect the anomaly by generating the normal profile of SIP messages as a probability distribution.

Ryu et al. [20] 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. Mehic et al. [21] 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).

8. Conclusion and Future Work

In this paper, we propose an effective scheme to detect and prevent SIP flooding attack. Not only can our scheme detect the flooding anomalies, but also find out the attackers to alleviate their adverse effects. To achieve this goal, we propose a versatile scheme by exploiting two-tier PFilter, a similar data structure as CBF. We demonstrate our PFilter in tier 1 is capable of filtering out a large portion of normal SIP messages. Then in tier 2, we take our collapse strategy in tier 2 to find out the attackers with great accuracy. Through extensive experiments, our scheme is demonstrated to gain great properties including effectiveness, efficiency, scalability.

PFilter can be extended to detect flooding attacks in other fields. In the near future, we will extend our detection schemes to more fields to demonstrate its strengths. For example, RTP flooding attack is another issue that is also troubling VoLTE. By applying our scheme in RTP flooding detection, we believe it could also function well and achieve the above merits.

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 and NSFC No. 61672350.

References

[1] H. Kim, D. Kim, M. Kwon, H. Han, Y. Jang, D. Han, T. Kim, and Y. Kim, "Breaking and fixing volte: Exploiting hidden data channels and mis-implementations," Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp.328–339, ACM, 2015.

[2] C.-Y. Li, G.-H. Tu, C. Peng, Z. Yuan, Y. Li, S. Lu, and X. Wang, "Insecurity of voice solution volte in lte mobile networks," Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp.316–327, ACM, 2015.

[3] G.-H. Tu, C.-Y. Li, C. Peng, and S. Lu, "How voice call technology poses security threats in 4g lte networks," Communications and Network Security (CNS), 2015 IEEE Conference on, pp.442–450, IEEE, 2015.

[4] J. Tang, Y. Cheng, and Y. Hao, "Detection and prevention of sip flooding attacks in voice over ip networks," INFOCOM, 2012 Proceedings IEEE, pp.1161–1169, IEEE, 2012.

[5] J. Tang, Y. Cheng, Y. Hao, and W. Song, "Sip flooding attack detection with a multi-dimensional sketch design," IEEE Transactions on Dependable and Secure Computing, vol.11, no.6, pp.582–595, 2014.

[6] H. Sengar, H. Wang, D. Wijesekera, and S. Jajodia, "Fast detection of denial-of-service attacks on ip telephony," 2006 14th IEEE International Workshop on Quality of Service, pp.199–208, IEEE, 2006.

[7] H. Sengar, H. Wang, D. Wijesekera, and S. Jajodia, "Detecting voip floods using the hellinger distance," IEEE transactions on parallel and distributed systems, vol.19, no.6, pp.794–805, 2008.

[8] A. Kumar and P.S. Tilagam, "A novel approach for evaluating and detecting low rate sip flooding attack," International Journal of Computer Applications, vol.26, no.1, pp.31–36, 2011.

[9] I. Hussain, S. Djahel, Z. Zhang, and F. Naït-Abdesselam, "A comprehensive study of flooding attack consequences and countermeasures in session initiation protocol (sip)," Security and Communication Networks, vol.8, no.18, pp.4436–4451, 2015.

[10] L. Fan, P. Cao, J. Almeida, and A.Z. Broder, "Summary cache: a scalable wide-area web cache sharing protocol," IEEE/ACM Transactions on Networking (TON), vol.8, no.3, pp.281–293, 2000.

[11] W. Meng, W. Li, and L.-F. Kwok, "Efm: Enhancing the performance of signature-based network intrusion detection systems using enhanced filter mechanism," Computers & Security, vol.43, pp.189–204, 2014.

[12] B.-H. Roh, J.W. Kim, K.-Y. Ryu, and J.-T. Ryu, "A whitelist-based countermeasure scheme using a bloom filter against sip flooding attacks," Computers & Security, vol.37, pp.46–61, 2013.
[13] D. Geneiatakis, N. Vrakas, and C. Lambrinoudakis, "Performance evaluation of a flooding detection mechanism for voip networks," 2009 16th International Conference on Systems, Signals and Image Processing, pp.1–5, IEEE, 2009.
[14] D. Geneiatakis, N. Vrakas, and C. Lambrinoudakis, "Utilizing bloom filters for detecting flooding attacks against sip based services," Computers & Security, vol.28, no.7, pp.578–591, 2009.
[15] Y. Rebahi, M. Sher, and T. Magedanz, "Detecting flooding attacks against ip multimedia subsystem (ims) networks," 2008 IEEE/ACS International Conference on Computer Systems and Applications, pp.848–851, IEEE, 2008.
[16] T. Ehrenkranz and J. Li, "On the state of ip spoofing defense," ACM Transactions on Internet Technology (TOIT), vol.9, no.2, pp.1–29, 2009.
[17] J. Tang and Y. Cheng, "Quick detection of stealthy sip flooding attacks in voip networks," 2011 IEEE International Conference on Communications (ICC), pp.1–5, IEEE, 2011.
[18] A. Akbar, S.M. Basha, and S.A. Sattar, "Leveraging the sip load balancer to detect and mitigate ddos attacks," Green Computing and Internet of Things (ICGCIoT), 2015 International Conference on, pp.1204–1208, IEEE, 2015.
[19] D. Golait and N. Hubballi, "Voipfd: Voice over ip flooding detection," Communication (NCC), 2016 Twenty Second National Conference on, pp.1–6, IEEE, 2016.
[20] J.-T. Ryu, B.-H. Roh, and K.-Y. Ryu, "Detection of sip flooding attacks based on the upper bound of the possible number of sip messages," KSII Transactions on Internet & Information Systems, vol.3, no.5, 2009.
[21] M. Mehić, M. Mikulec, M. Voznak, and L. Kapicak, "Creating covert channel using sip," International Conference on Multimedia Communications, Services and Security, vol.429, pp.182–192, Springer, 2014.

Na Ruan received the B.S. degree in Information Engineering and the M.S. degree in Communication and Information System from China University of Mining and Technology in 2007 and 2009 respectively. She received D.E. degree from the Faculty of Engineering, Kyushu University, Japan in 2012. Since 2013, she joined the Department of Computer Science and Engineering of Shanghai Jiaotong University as Assistant Professor. Her current research interests are in wireless network security and game theory. Dr. Ruan is a member of the Information Processing Society of Japan (IPSJ), China Computer Federation (CCF), ACM and IEEE.

Mingli Wu received his B.S. degree from Nanchang University in 2014. He is currently a master student in Shanghai Jiao Tong University. His research interests include network security and smartphone privacy.

Shiheng Ma received his B.S. degree from Nankai University in 2013. He is currently a Ph.D. student in Shanghai Jiao Tong University. His research interests include network security.

Haojin Zhu is currently an Associate Professor with Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. He received his B.Sc. degree (2002) from Wuhan University (China), his M.Sc. (2005) degree from Shanghai Jiao Tong University (China), both in computer science and the Ph.D. in Electrical and Computer Engineering from the University of Waterloo (Canada), in 2009. His current research interests include wireless network security and distributed system security. He is a member of IEEE.

Weijia Jia is currently a Zhiyuan Chair Professor in the Department of Computer Science and Engineering at Shanghai Jiao Tong University. Professor Jia received PhD in Computer Science from Polytechnic Faculty of Mons, Belgium in 1993. He joined German National Research Center for Information Science (GMD) in Bonn (St. Augustine) from 1993 to 1995 as a research fellow. From 1995 to 2013, he joined Department of Computer Science, City University of Hong Kong as an assistant/associate and full professor. His research interests include next generation wireless communication, protocols and heterogeneous networks; distributed systems, multicast and anycast QoS routing protocols.

Songyang Wu is an associate professor at The Third Research Institute of Ministry of Public Security, China. Vice director. He received his Ph.D. Degree in computer Science from TongJi University, China in 2011. His current research interests are in information security, cloud computing and digital forensics.