Proactive detection of DDOS attacks in Publish-Subscribe networks

PROACTIVE DETECTION OF DDOS ATTACKS IN PUBLISH-SUBSCRIBE NETWORKS

Bander Alzahrani[1], Vassilios Vassilakis[2], Mohammed Alreshoodi[3], Fawaz Alarfaj[4] and Ahmed Alhindi[5]

[1]Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
[2]School of Computing, Engineering and Mathematics, University of Brighton, Brighton, United Kingdom
[3]College of Computer, Qassim University, Buraydah, Saudi Arbaia
[4]Al Imam Mohammad Ibn Saud Islamic University (IMSIU), Saudi Arabia
[5]College of Computers and Information Systems, Umm Al-Qura University, Makkah, Saudi Arabia

ABSTRACT

Abstract. Information centric networking (ICN) using architectures such as Publish-Subscribe Internet Routing Paradigm (PSIRP) or Publish-Subscribe Internet Technology (PURSUIT) has been proposed as an important candidate for the Internet of the future. ICN is an emerging research area that proposes a transformation of the current host centric Internet architecture into an architecture where information items are of primary importance. This change allows network functions such as routing and locating to be optimized based on the information items themselves. The Bloom filter based content delivery is a source-routing scheme that is used in the PSIRP/PURSUIT architectures. Although this mechanism solves many issues of today's Internet such as the growth of the routing table and the scalability problems, it is vulnerable to distributed denial-of-service (DDoS) attacks. In this paper, we present a new content delivery scheme that has the advantages of Bloom filter based approach while at the same time being able to prevent DDoS attacks on the forwarding mechanism. Our security analysis suggests that with the proposed approach, the forwarding plane is able to resist attacks such as DDoS with very high probability.

KEYWORDS

Distributed denial-of-service attack; information centric network; Bloom filter.

1. INTRODUCTION

Today's Internet architecture was deployed in the 1960/70's to address the basic communication needs that were necessary at that period. And indeed, it has successfully and efficiently solved the technical problems that arose in the past decades. In fact, a key aim of the original architecture was to interconnect end-to-end limited machines that were used by a small number of trusted users to allow remote access and share a small amount of data, at a time when the network speed was just a few Kbps. Therefore, early usage was mainly around host-to-host services such as remote access and file transfer. However, this is radically different from the situation today where security threats such as viruses, phishing and distributed denial-of-service (DDoS) attacks are widespread, the number of Internet users is in rapid increase (≈ 3 billions [24]) and there is an increasing need to support emerging mobile applications at large scale. The Internet cannot efficiently handle these emerging issues and it has been difficult to apply major architectural

DOI: 10.5121/ijnsa.2016.8401

changes to the Internet to cope effectively with these requirements [25]. Subsequently, a number of important shortcomings of this architecture have been reported [23,16]. In fact, the Internet is not working efficiently when it comes to security, routing scalability, QoS, and content delivery.

From a security perspective, the current Internet architecture was built with the principle that it would be used by trustworthy users in a cooperative and open environment. Therefore, user privacy, information integrity, and user authentication mechanisms were not requirements and consequently have not been considered in the original design. Moreover, since the original design is based on named hosts; it provides the sender with a complete control over the communication to send almost any arbitrary traffic at any time to anyone and without any pre-authorisation, and the Internet is highly suitable to deliver this traffic to the intended destination. Unfortunately, the openness of the Internet, which was a critical success factor, now offers no protection for network nodes. Therefore, it is relatively easy to launch DDoS attacks which may cause severe and serious issues, e.g. by flooding some limited resources on the Internet, thus preventing legitimate network traffic. It is also relatively easy for malicious users to hide their identity by using a fake source address.

To mitigate DDoS attacks, a number of network capability proposals have been deployed with various success [32]. Network capability refers to any mechanism that allows routers to statelessly verify if the packet has been previously authorized [8]. In such a proposal, each packet carries information so that routers can identify if the packet is requested by the receiver. Therefore, any sender must first get an authorization-to-send capability from the receiver before being able to send data. The receiver permits the communication either directly [46,45], or indirectly using a third party system [44,35]. Source address filtering [17] is an example of network capabilities that enables routers to filter traffic at the network ingress and egress by identifying spoofed IP address. Although it prevents attackers from replacing their IP address with a random fake one, it is not effective in today's network where attackers can use a large number of compromised machines, i.e. zombies, with a real IP address.

As these issues are related to the core architecture of the Internet, most of the solutions are just add-on functionalities, which in turn have increased complexity. A number of research discussions have investigated the current challenges to identify the main requirements of a future Internet. In this context, a new networking paradigm, termed information-centric networking (ICN), has recently been proposed, aiming at replacing the existing host-centric networking paradigm to address the aforementioned issues [13, 3]. One of the basic features of ICN is to name information items themselves, rather than naming and connecting hosts. This fundamental shift has direct implications on network and systems security issues such as DDoS, which is the focus of this work.

There are some research projects that have developed advanced ICN solutions, such as: Data Oriented Network Architecture (DONA) [29]; Named Data Networking (NDN) [27,26]; Scalable and Adaptive Internet Solutions (SAIL) [2,11]; Content Mediator Architecture for Content Aware Networks (COMET) [1]; and, Publish-Subscribe Internet Technology (PURSUIT) [18,31].

The forwarding mechanisms, used in above mentioned ICN architectures, can be broadly classified into two types: stateful (e.g. as in NDN) and stateless (e.g. as in PSIRP/PURSUIT). The problem of DDoS attacks in ICN architectures that use stateful forwarding has been extensively studied in several works (e.g., [22,14]). Therefore, in this work we study the implications of DDoS attacks on stateless forwarding, using as an example the PSIRP/PURSUIT ICN architecture (in the following referred to as PURSUIT architecture). The PURSUIT architecture defines three distinct types of network entities, namely publishers, subscribers, and the mediation system [15]. Publishers announce and provide the available information objects by sending in the

network the publication messages. Subscribers send the subscription messages whereby showing their interest in receiving certain information objects. Contrary to the traditional host-centric paradigm, the subscriber does not need to be aware of the publisher's IP address. The mediation system consists of two logical entities: the rendezvous function (RV) [34,43] and the topology management (TM) function [21,36]. These two entities control and manage another entity: the forwarding (FW) function [28,37,39,47,40,38]. Each information item in the network is uniquely identified by the pair rendezvous identifier (RId) and the scope identifier (SId). SIds are used to organise the information items into scopes, whereas RIds identify items within scopes. The RV is responsible for receiving the publication and subscription messages and determining the set of publishers and subscribers for a given RId of the information item within a specified scope. When the RV detects a match, it contacts the TM who is responsible for maintaining the intra-domain topological knowledge within a domain and for constructing the delivery path from the publisher to the subscriber. When the path is ready, the content can be delivered via a chain of FW nodes from the publisher to the subscriber by deciding where packets should be forwarded.

Although the communication paradigm of PURSUIT has tried to decouple the information from its source location, so that a sender would not be able to send anything unless a subscription is received, recent studies have shown that a DDoS attacks are still possible [9]. In [37], it has been shown that the Line Speed Publish/Subscribe Inter-networking (LIPSIN) forwarding scheme used in PURSUIT is vulnerable to computational attacks and replay attacks which can be used to launch DDoS attacks. In the same work, in order to offer DDoS resistant forwarding, the zFormation technique has been proposed. This technique has been studied from the viewpoint of TM scalability in [7]. The works of [37, 7] have been extended in [4] to mitigate the brute-force attacks on the LIPSIN scheme and to reduce the DDoS attack probability.

Building upon the LIPSIN scheme and network capabilities work [8,17,37], this paper proposes a content delivery approach that effectively prevents DDoS attacks using network capabilities. The rest of the paper is organised as follows. In Section 2, we present an overview of the PURSUIT ICN architecture, the scheme used for packet forwarding and its security vulnerabilities. We also point out the limitation of the existing solutions. In Section 3, we present our proposed secure forwarding mechanism. In Section 4, we analyse the resistance of our solution to DDoS, and conclude the paper in Section 5.

2. BACKGROUND AND RELATED WORK

2.1. The PURSUIT architecture

The PURSUIT project [19] brings two concepts to form an ICN solution: mediated assisted forwarding and publish/subscribe communication. The basic PURSUIT architectural model is given in Fig. 1. In order for a publisher to announce a publication, it must know the SId within which the information object to be published, as well as to create an RId for the item. As we observe in the figure, initially the publisher notifies the RV that it can provide an information item, by sending the corresponding RId and SId. Next, the subscriber notifies RV that it wishes to receive the item. The RV detects the matching and asks the topology manager to compute the delivery path from the publisher to the subscriber. The TM, after having accomplished this task, encodes the delivery path into a Bloom filter [10] and sends it to the publisher. Finally, the publisher and the on-path FW nodes will use this Bloom filter as the forwarding identifier (FId).
Below we will explain how the FId is created and used for packet forwarding decisions. LIPSIN [28] is one of the developed forwarding techniques of the PURSUIT ICN and uses the Bloom filter based content delivery. Bloom filters are memory-efficient data structures that perform fast

creation and membership tests. In order to realise content delivery using Blooms, each network link is identified with a unique link identifier (LId). Typically, an LId is an *m*-bit array

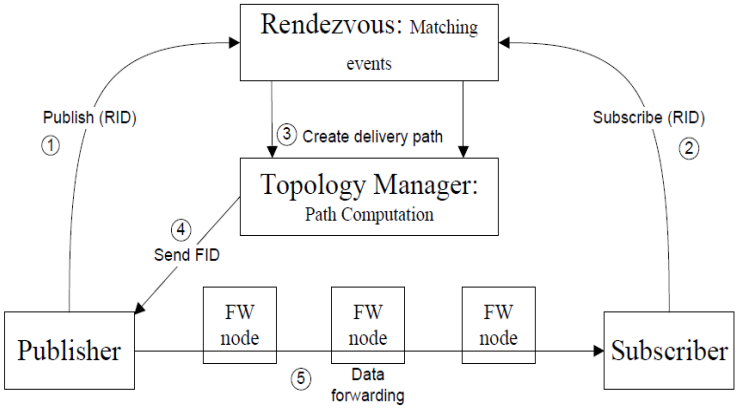

Figure.1. A general view of the PURSUIT architecture.

and *k* hash functions are used to set bits to 1, with $k << m$. These two parameters value are subject to some optimization criteria [12]. Some typical values are $m = 256$ and $k = 5$ [28,37]. An example of the Bloom-filter based approach is illustrated in Fig. 2. Consider a small network consisting of 5 forwarding nodes, one publisher and 3 subscribers. Each link is assigned a statistically unique LId with $m = 8$. Assume that Sub-B wishes to receive an information item from Pub-A. For this to be achieved, as discussed above, the TM has to encode the LIds of the delivery tree into an FId and send it to Pub-A. The encoding is done by performing the bitwise OR operation. As shown in Fig. 2, Pub-A places the FId in the header and forward the packet to its edge forwarding node (FW-1). Then, this node examines each of its outgoing interfaces (i.e. LId-2 and LId-3), by performing the forwarding membership test. The forwarding test is done by performing the logical AND operation on the FId and the outgoing LId. If the result of this AND operation equals to the LId, then the packet will be sent via the corresponding link. As we observe in Fig. 2, the packet will be sent to the FW-3. This FW node will also perform the same forwarding test on all its outgoing links, and so on. Finally, the requested information item will reach Sub-B, but not Sub-A and Sub-C, since the forwarding tests at FW-1 and FW-5 over LId-2 and LId-8, respectively, will fail.

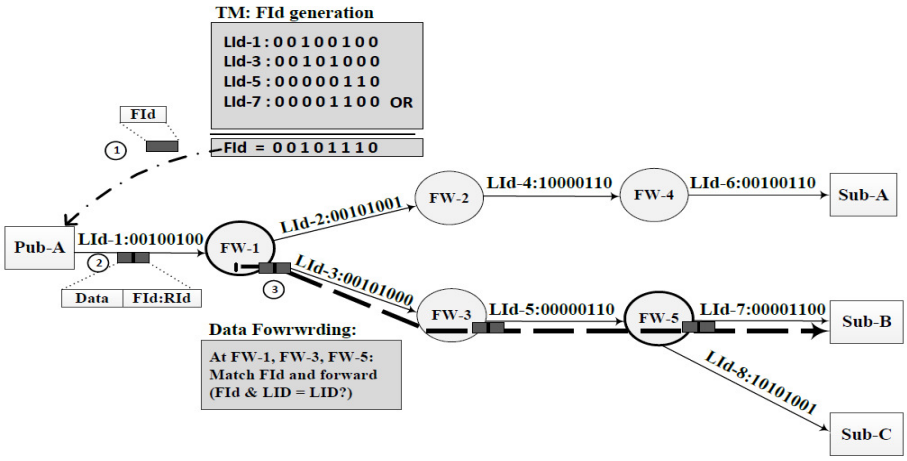

Figure.2. LIPSIN based forwarding using Bloom filters.

2.2 False positive issue and security chain reactions

One limitation of the LIPSIN forwarding is that some packets may be unnecessarily forwarded to non-provisioned links. That is, the forwarding test, described above, may give false positives. For instance, suppose that LId-8 is equal to 00101000 in Fig. 2. Since this LId has not been included in the path from Pub-A to Sub-B, the FId created by the TM is as shown in the figure. Nevertheless, the forwarding test at the FW-5 will falsely produce a positive result for the LId-8. This shortcoming may generate infinite loops which become a security issue if exploited by a malicious node. It may lead to DDoS attacks against end-users or some parts in the network by deliberately creating false positives in sensitive links. In practice, the attacker could perform several attacks such as flow duplication, packet aggregations, and forwarding loops, overwhelming the network resources [40].

2.3 Security vulnerabilities in the forwarding plane

In this section, we highlight the vulnerabilities of the LIPSIN mechanism. First of all, note that when all bits are set to 1 in a FId, the packet will be delivered to every network node, because the forwarding test will give a positive result at every FW node. To address this issue, LIPSIN sets an upper limit on the number of bits set to 1 in a FId. The ratio of the bits set to 1 over the total bits, m, is called fill factor and denoted by ρ. In the example of Fig. 2, the FId has $\rho = 4/8 = 0.5$. In the literature, a typical value of the fill factor upper limit is $\rho_m = 0.5$ [37,40]. Any FId with $\rho > \rho_m$ will be immediately rejected by FW nodes.

Another security threat that may arise is when the attacker is exploiting false positives and launching the brute-force attack to reach a targeted end-node. To achieve that, the attacker tires all possible FIds with the aim to obtain a valid FId that reaches a target node. The number of possible FIds depends on the FId length, m, and the maximum fill factor, ρ_m. The probability, p_{fw}, to guess a working FId created with a maximum fill factor of ρ_m, k hash functions and goes to a target l hops away can be calculated as follows [37,4]:

$$p_{fw} = \rho_m^{k \cdot l} \tag{1}$$

Other types of attacks, such as replay attacks and computational attacks, are also possible. In a replay attack, some non-requested traffic with a previously created valid FId, is injected into the network. A computational attack is started by gathering legitimate FIds and inspecting the correlation of their bit patterns.

2.4 Secure zFormation

One factor that strengthens the security threats is the use of time-invariant LIds. The zFormation approach, proposed in [37], makes FIds expire after a certain time by periodically changing the LIds every Δt (duration of time that a LId is valid). A cryptographically secure hash function, called zFormation, computes the LIds based on the following parameters:

- Some content-based information, T, such as the flow Id or the RId.
- Periodically changing time-based secret key shared with FW nodes, $K(t)$.
- The port number of an incoming or outgoing link, I and O, respectively.
- The optimization index, d, to mitigate the false positives proposed in [28].

As a result, the LIds become dynamic and bound to a specific content-related information and have limited time duration. This approach provides sufficient protection against replay attacks

and computational attacks. However, in [4], we have demonstrated that the zFormation solution is still vulnerable to successful DDoS attacks if either the fill factor of the FId is too large or Δt is too long. In practice, we found that an attacker can inject unwanted traffic, using a fake FId, in a path up to 5 hops within only 23s, using a reasonable attacking rate of 10^6 packet/s, i.e. by having a packet size of 10^3 bits and 10^9 bit/s capacity on each edge link. The issue of brute-force attacks has been addressed in [4,5], where the impact of ρ_m and m, respectively, on the attack capability has been studied.

Moreover, although the zFormation mechanism has effectively made DDoS attack more expensive to be successfully launched, in [9] it has been shown that attacker can penetrate the secrecy of LIds. Consequently, they could launch DDoS to a specific target if a sufficient number of LIds are identified. In this case, the attacker launches what is called a reverse-engineering attack over the secret LIds and by using a botnet that is randomly distributed around the network. In particular, all compromised nodes in the botnet subscribe to information items from each other. Assuming that the attacker has knowledge of the network topology, it first collects all legitimate FIds created by the TM for all pairs of bot nodes. Since the network topology is known, the attacker is able to know the connectivity graph and the FW nodes of which each FId passes through. Therefore, for each FW node it can figure out the LId by computing the intersection of the FIds that passes through the FW node by taking a bitwise AND operation over the FIds. The accuracy of the computed LIds may just be approximation to the real ones with some extra bit set to one, but this does not affect the attack, unless the maximum fill factor is exceeded. The attacker then can build FIds to attack a victim without receiving any help from the TM.

However, it is obvious that after obtaining the LIds and building a valid FId, the attack is only limited to a short time duration due to the periodic LIds/FIds updates. Therefore, a frequent update of the LIds can minimise the effect of the attack, since the old FIds are expired after the update.

3. THE PROPOSED SECURE FORWARDING MECHANISM

3.1 Overview

To mitigate brute-force attacks and subsequently prevent DDoS attacks, a validation mechanism is required to check the legitimacy of FIds that are sent by publishers to the network. This mechanism should not add any additional overhead on the TM, such as maintaining a large number of shared keys with each FW node in the network. In practice, we seek a mechanism that prevents the attack locally on the forwarding plane in a stateless manner and without involving any other entity rather than the corresponding FW node. We consider these issues in the proposed secure forwarding mechanism. In this scheme, in addition to performing the forwarding test at each FW node, as proposed in the zFormation technique, we also introduce a security test at the ingress node, so that any attack is prevented at early stage. In the following, the FW node that is directly attached to a user node (publisher/subscriber) is called Edge-FW. For instance, in Fig. 2, the Edge-FW nodes are FW-1, FW-4 and FW-5. The approach proposed in this paper uses pre-defined and fixed LIds, as in the traditional LIPSIN mechanism. This is different from is the zFormation technique, discussed in Section 2.4, which uses dynamic changeable LIds.

Our solution is based on the following assumptions: (1) links from users to Edge-FW nodes are assumed to be local, i.e. not used by the TM in the FId generation. On the other hand, links from Edge-FW node to users are global and are included in FId generation. (2) Each Edge-FW node maintains two 128-bit long master keys, $k1$ and k_2. (3) As in current access network scenarios, the attacker has no special privileges in the network but can send arbitrary packets up to the capacity of the access link.

3.2. Secure FId generation

We should recall that the TM knows the network topology. Also, it is mainly responsible for finding the best path between publishers and subscribers. Our proposed forwarding scheme is illustrated by a simple example of Fig. 3 (a). In this example, FW-1 is the Edge-FW node responsible for performing the security check. We refer to the legitimate forwarding identifier contracted by topology manager as FId and to its encrypted version as eFId; whereas the one utilised by the publisher is denoted as $eFId_p$ and its decrypted version is FId_p. The hash that is performed over eFId is called *h*, and the hash used by the publisher is called hp. When a legitimate, non-hostile publisher is using the network, each pair of each pair of $\{FId,FId_p\}$, $\{eFId,eFId_p\}$, and $\{h,h_p\}$ will be identical. This is because a legitimate publisher will be using eFId that is generated by the Edge-FW, without any form of malicious manipulation.

In our proposed scheme, when generating FId, link from publisher to Edge-FW node is not included in the delivery tree. Also, TM provides Edge-FW node with an outgoing interface number or an access media address to the publisher where FId should be forwarded. Therefore, in Fig. 3, to create FId that connects the publisher Pub with the subscriber Sub, TM encodes the

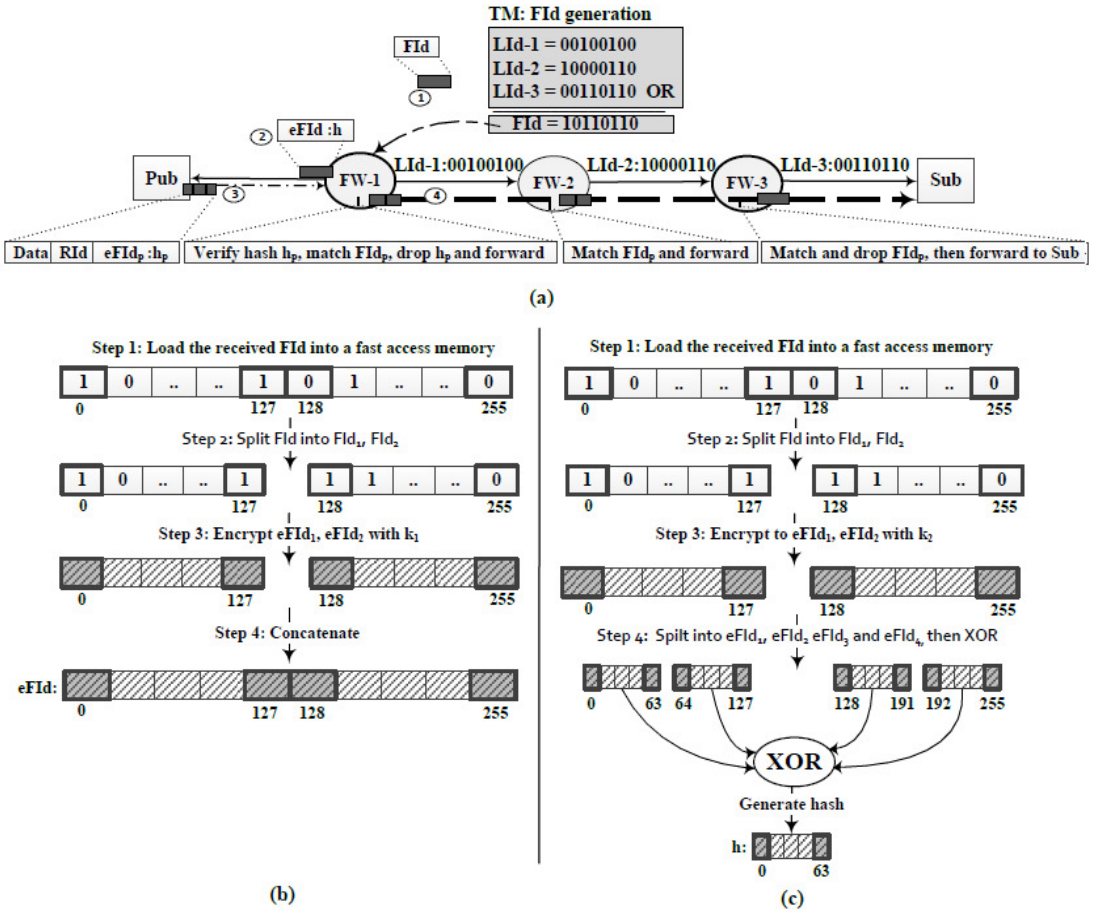

Figure.3. Edge-FW approach: (a) An example of the FId creation and packet forwarding. (b) Encrypting the FId at the Edge-FW node. (c) Generating the hash h at the Edge-FW node.

links LId-1, LId-2, and LId-3 into the Bloom filter and then performs a bitwise-OR operation over them. FId generated in this way is then sent to FW-1 node which is the publisher's Edge-FW node.

Upon receiving the FId by the FW-1, the AES algorithm is applied to encrypt it. The encrypted version is denoted as eFId. This is done in order to preserve the confidentiality of the FId. This way, computational attack by a potentially malicious publisher can be prevented. It should be noted that in this scheme, all encryption and decryption processes are performed using AES. This encryption algorithm can be implemented in a number of efficient ways, and it was designed from the very beginning as a cipher which can be implemented efficiently, both in software or hardware. It is considered scalable and performs well when used in IPsec/IPv6 [20]. Recently, it has been shown that AES can be accelerated to work at line-speed of 10 Gbps requiring only 8 processor cores running at 3 GHz [30]. This feature makes AES particularly suitable for PURSUIT ICN.

To prevent DDoS attacks, the FW-1 generates a 64-bit hash h over the eFId. This way, the resultant hash becomes bound to a certain identifier. Afterwards, the pair $\{eFId,h\}$ is sent to the Pub and will be used to send traffic to the Sub. The procedure of encrypting the FId is shown in Fig. 3(b). To encrypt the received FId, FW-1 copies FId into fast memory and then splits it into two parts: FId1, FId2, each of which is 128-bit long (for compatibility with the keys length. Next, each part is encrypted separately with the master key k1 and results in eFId1, eFId2. The last step is performed by concatenating eFId1 and eFId2 to form a 256-bit encrypted eFId. The hash creation procedure is shown in Fig. 3(c), where the process starts by repeating Steps 1 and 2 of the FId encryption procedure. In Step 3, FW-1 encrypts FId_1 and FId_2 with k_2 which protects the hash.

Next, in Step 4, these sub eFIds are further split into four parts, each of which is 64bits long. Then, a bitwise-XOR operation is taken over them to form a 64-bit hash h. The purpose of this process is to preserve the integrity of the original eFId so that any forged eFIdp, used by the publisher, is identified.

3.3 Secure FId forwarding

When the pair $\{eFId,h\}$ is created, it is then forwarded by FW-1, as shown in Fig. 3, through the outgoing interface given by TM to the relevant directly connected publisher. The publisher places this pair on each transmitted packet. Next, when the FW-1 node receives a packet, initially it will perform two tests: the security test and the forwarding test. The security test will validate the received eFIdp (i.e., whether this eFIdp is legitimate). This test is done at the Edge-FW node and only for packets received from the publisher's side. The forwarding test, as described previously, is the basic LIPSIN membership check and will decide the next hops of the received packets. A packet is sent to next hop only if it can pass both tests. In the security test, the Edge-FW node also verifies the integrity of the received eFIdp. This verification is performed as shown in Algorithm 1.

If the packet will pass the security test, the eFIdp is considered legitimate. Then, the Edge-FW node drops the hash h and replaces the eFIdp with the FIdp, which should be identical to the original FId in a non-malicious communication. Afterwards, the forwarding test is done for each outgoing interface using FIdp. If the test is positive, the packet will be sent to the next FW node in the path. The forwarding test is done at each FW node until the packet reaches the Edge-FW node of the subscriber. At this stage, the FIdp is discarded before forwarding the packet to the subscriber. This prevents the reverse-engineering attack mentioned in Section 2.4, where the attacker tries to obtain simultaneous valid plaintext forwarding identifiers from compromised subscribers to obtain useful information about the network topology by analysing the correlation between a large number of forwarding identifiers.

According to our proposed approach, the master key k_2, is periodically changed to mitigate replay attacks. Therefore, an attacker who possesses a valid encrypted pair {eFId,h} cannot use it maliciously to send unwanted traffic forever, as the hash is expires once the key is changed. The attacker is limited within the time window of k_2 update. For long-lived flows, FIds that are about to expire can be updated when TM send new k_2 keys to Edge-FW nodes. Also, this approach requires that RV checks periodically its event table periodically within the specified time window looking for events that need to be updated. Hence, the relevant Edge-FW node is informed by TM about these events in order for the FId to be updated with a new hash. If an information item is not wanted any longer, the subscriber will send a remove subscription request. Otherwise, RV will assume that the subscriber is still interested in the item and the Edge-FW node will be informed about extending the communication by generating a new hash. The problem of selecting a suitable time window for the aforementioned FId update operation has been studied in [6].

Input: A pair of $\{eFId_p, h_p\}$ extracted from the incoming packet, with lengths of 256-bit and 64-bit respectively.

1 **Begin**
2 Split $eFId_p$ into $eFId_{p1}$ and $eFId_{p2}$, each of which is 128-bit long.
3 Decrypt $eFId_{p1}$ and $eFId_{p2}$ to FId_{p1} and FId_{p2}, respectively, using k_1.
4 **if** $\rho \leq \rho_m$ **then**
5 Encrypt FId_{p1} and FId_{p2} to $eFId_{p1}$ and $eFId_{p2}$ using k_2.
6 Split $eFId_{p1}$, $eFId_{p2}$ into four parts.
7 Perfom a bitwise-XOR operation over the generated parts to create a 64-bit hash h.
8 **if** $h_p = h$ **then**
9 The $eFId_p$ has passed the security check, so combine the FId_{p1} and FId_{p2} found in 3 and perform the LIPSIN forwarding check.
10 **else**
11 Drop the packet and count a failure attempt against the publisher.
12 **End**

Algorithm 1. The security test procedure at the Edge-FW mode

4. ATTACK ANALYSIS AND DISCUSSION

Although most of the proposed false positive reduction techniques [28,12,42] have effectively added a control over the false positive rate and kept it at a minimum, they are unfortunately not useful when brute-force attacks are considered. These proposals aim at optimizing FIds in terms of the false positive rate, at their formation stage and then placing some relevant optimization parameters in the packet to be used by the forwarding plane. However, these techniques do not consider anomalous behavior where publishers maliciously ignore or alter the in-packet parameters; consequently, an attacker subverts the forwarding process such that the minimal false positive rate no longer applies. For example, the LId-Tag (LIT)-based solution proposed in [28] aims at reducing the false positives. According to this approach, a subset of LIds that results in lowest false positive rate is chosen from the set of all available d LIds. Then, the d value corresponding to the selected subset is inserted in the packet header. This strategy is not suitable for mitigating brute-force DDoS attacks, since the attacker is able to use an arbitrary value of d to achieve a successful attack with high probability. Also, taking into account that

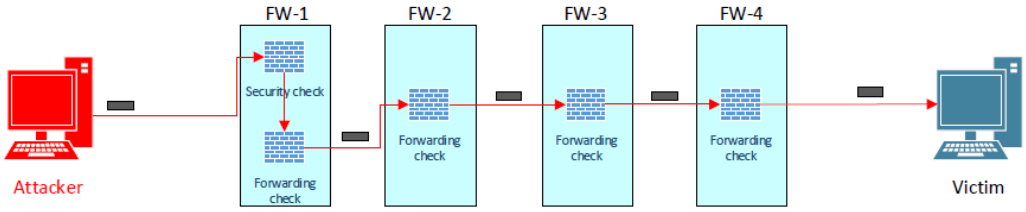

Figure.4. The sequential checks to be achieved by the attacker to reach a victim 4-hop away from an attacking node.

this strategy does not perform a verification mechanism at the FW node to force the attacker to use the original selected value of d. Indeed, it would not be possible for the FW node to know which value of d is actually the correct one.

On the contrary, our proposed secure forwarding mechanism can effectively stop the attacks described in Section 2.3. For example, if an attacker wants to inject unwanted traffic to a victim who is four hops away, then he first must pass both the security test and the forwarding test at his Edge-FW node, and also pass three forwarding tests in the subsequent in-path FW nodes, as shown in Fig. 4. Below, we perform an analysis of the probability of injecting unwanted traffic using the brute-force method.

The probability of passing the forwarding test p_{fw}: This is the probability of having false positives in the delivery path. This is calculated as in (1).

The probability of passing the security test p_{sc}: Introducing the security test at the Edge-FW node makes DDoS attack using brute force much expensive. This is because for this type of attack, the attacker has to guess the hash correctly. Therefore, for example, if the hash is 64-bit long, the attacker has to send 2^{64} packets to the Edge-FW node, and the probability of being successful is approximately 5.4×10^{-20}. This is a very low probability which makes this attack scenario unrealistic in practice.

However, a better strategy the attacker could adopt is to launch the attack taking into account that fact that all hash functions suffer from the birthday paradox attack [41]. It is a generic collision-finding attack performed by using a probabilistic model to reduce the complexity of cracking a hash function. Let us demonstrate how a collision can be found in the context of our approach, let us assume H is a hash function such that $H : D \rightarrow R$, where D is the set of all possible combinations of *FIds*, R is the range of H, and |R| = r. A hash collision occurs when having distinct $eFId_1$, $eFId_2 \in D$ where $H(FId1) = H(FId2)$. By exploiting the birthday paradox, the attacker creates a packet with an independently and uniformly random selected FId_1, ..., FId_q, along with a random hash h_1,..., h_q, where q is the number of the attempts, to be sent to the Edge-FW node. The aim is to cause a hash collision at the Edge-FW node where two distinct FIds give the same hash; so, the packet is treated as legitimate and, hence, successfully passes the security test. This strategy significantly reduces the number of attempts q, as it is expected that the attacker finds a collision within $r^{1/2}$ attempts with a probability of $psc \approx 0.5$ [41].

$$x \approx \sqrt{2r \ln \frac{1}{1 - p_{sc}}} \tag{2}$$

where r is the number of all possible hash outputs.

The number of required attempts increases as the hash length increases, and generally, the hash function length is chosen to be large enough to make the number of attempts infeasible. However, since the Edge-FW approach utilizes a hash size of only 64-bit long, to limit the security overhead, this means that within $\approx 42 \times 10^8$ attempts the attacker may pass the security test with probability psc = 0.5. Therefore, we introduce a blocking mechanism where a publisher who exceeds a certain amount of packet drops is assumed to be a malicious user and, hence, is blocked. The packet drops at the Edge-FW node occur due to a failure in either the security test or the forwarding test. The blocking mechanism is an effective countermeasure, as it prevents the attack at its early stage. However, an attack success probability of 0.5 is considered high as the attacker might be lucky and pass the security test based on his first attempt so the blocking mechanism is not effective in this case. Therefore, the network designer may select a very low probability of success psc and then block any attack before reaching the required number of attempts of achieving the selected probability. To determine the number of required attack attempts x, that inject random pairs $\{FId,h\}$, to achieve a target probability psc of finding a hash collision, we adopt the approximation of [33].

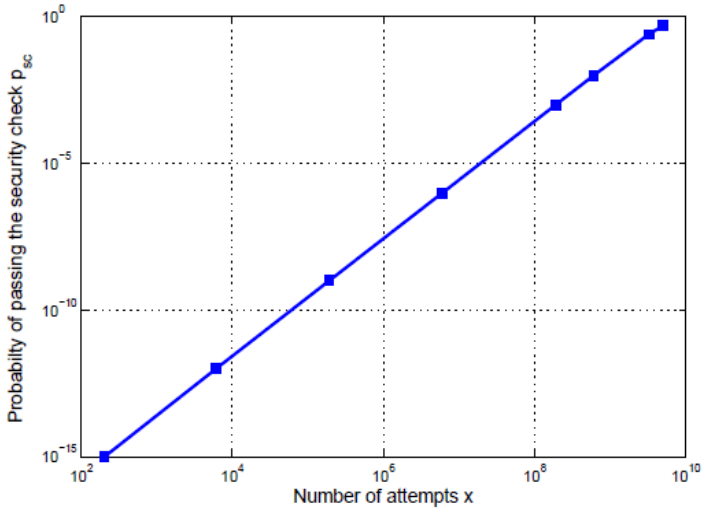

Figure.5. The expected number of attempts required to find a collision in 64-bit long hash for different probability p_r.

In Fig. 5 we show the number of required attempts, x, to find collision in a 64-bit long hash (that is $r = 1.8 \times 10^{19}$), assuming various different desired collision probabilities $p_{sc} = \{1\times10^{-15},1\times10^{-12},1\times10^{-9},1\times10^{-6},1\times10^{-2},0.1,0.25,0.5\}$. We observe that the number of required attempts, x, is decreased as the selected p_{sc} decreases. For example, the attacker would need to generate and send approximately 6×10^8 packets to find a collision and pass the security test with a probability of 1%. However, if we assume even a lower probability of $p_{sc} = 10^{-9}$, this would require approximately 1.9×10^5 attempts. For this reason, we suggest blocking the publisher after making $\approx 10^5$ failed attempts, which is a large number and the probability that the attacker could succeed is very small and approximately equal to 10^{-9}. In fact, the probability of passing the EdgeFW node is actually lower than that because the attacker also has to pass the forwarding test.
Our approach also introduces an additional mitigation layer to brute-force attacks. This is because in this approach, the attacker loses the ability to inject maximally filled eFIdp to the network, which is considered a fundamental strategy to follow as it increases the probability of passing the forwarding test. This strategy now becomes harder to be used because the bit pattern of the injected eFIdp will be changed after the decryption process at the Edge-FW node, which may result in an FId with two cases: an FId with $\rho > \rho_m$, so the packet is dropped or an FId with the

chance of success is low. Hence, to reach a victim, the attacker must be" incredibly lucky" and pass both tests at the Edge-FW node and also the forwarding test at all in-path FW nodes. The probability of a successful attack can be calculated by:

$$p_a = p_{sc} \times p_{fw} \tag{3}$$

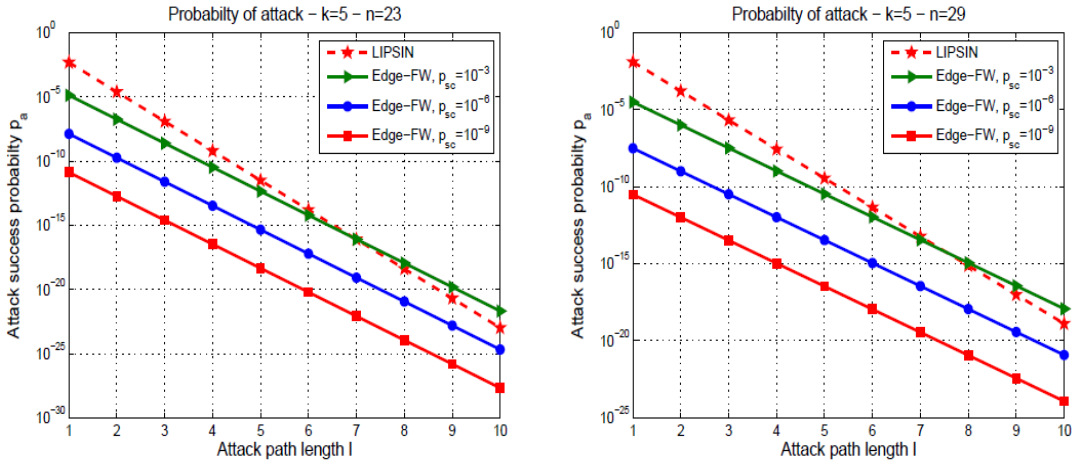

Figure.6. Comparison of the probability of successful attack for different path lengths between LIPSIN and Edge-FW approaches. The number of hash function used is $k = 5$ with $n = \{23, 29\}$. The Bloom filter size used in LIPSIN is $m = 320$ whereas in Edge-FW $m = 256$ plus 64-bit hash h.

This probability of successful attack pa is actually an upper bound, since this equation assumes that the decrypted forged FId is still maximally filled.

Figure 6 depicts the probability pa for varying attack path lengths l for the current LIPSIN scheme and our proposed Edge-FW scheme. We compare the probability pa in LIPSIN using a Bloom filter size of $m = 320$-bit; whereas in our approach $m = 256$-bit is used. This is because the Edge-FW approach introduces an extra size of 64-bit hash, so the overall introduced size in the Edge-FW approach is 320-bit. In both approaches, we assume a network that supports different sizes of a maximum delivery tree, $n = \{23,29\}$ LIds, with $k = 5$. Subsequently, in the case of having 23 LIds, ρ_m would be 0.347 for the basic LIPSIN and 0.423 for the Edge-FW mechanism; and when using 29 LIds ρ_m would be 0.48 and 0.42, for the basic LIPSIN and the Edge-FW mechanisms, respectively.

The figure on the left corresponds to the scenario where $n = 23$. We observe substantial improvement of the probability pa for the Edge-FW approach. For instance, when $p_{sc} = 10^{-6}$, then $p_a \approx 1.3 \times 10^{-8}$. This is much lower than ≈ 0.0001 of the current LIPSIN scheme. The forwarding plane becomes even more secure when psc is chosen to be 10^{-9} as the probability of attack, pa, in this case sinks to $\approx 1.3 \times 10^{-11}$. It can be noticed that when p_{sc} is selected to be 10^{-3}, the Edge-FW approach does not perform well compared to the LIPSIN, especially when the path increases over 6 hops. This is true because the value of ρ_m used in the Edge-FW mechanism is significantly small compared to that in the LIPSIN. It is possible to increase the size of the maximum supported delivery tree as in the right figure where $n = 29$ at the cost of having a higher attack probability, p_a.

The mechanism introduced in this paper is important as it stops an attacker sending malicious packets using a random FId, which in the PURSUIT architecture could be a multicast tree and potentially could reach a large number of end-users that have not requested the data. It is useful to also compare this with contemporary IP networking where it is possible to send packets with spoofed IP addresses or packets without a user requesting them. The mechanism presented here stops this kind of malicious behaviour, which could be used to launch a DDoS attack. However, it is also worth noting that it does not stop all forms of DDoS attack, for example a large bot-net which simply requests many data items would not be stopped by this mechanism (although the load may be spread over multiple publishers without allowing the attackers to choose a specific publisher to overload, thus aiding existing load-balancing approaches to DDoS).

5. CONCLUSION

In this paper we propose a novel secure Bloom filter based forwarding mechanism for ICN. We have demonstrated that this approach can provide good protection against DDoS attacks. The forwarding mechanism uses network capabilities to identify illegitimate traffic, via its forwarding identifier, at the ingress nodes. Using our approach, the probability of a successful a DDoS attack can be substantially reduced in comparison to existing schemes, to the point where an attack is unrealistic. Moreover, an attacker can be easily detected and blocked at the very early stages of the attack. Furthermore, the proposed scheme does not require storing forwarding state at the Edge-FW nodes in order to detect any in-progress attacks.

REFERENCES

[1] COntent mediator architecture for content-aware networks (COMET). www.comet-project.org, accessed on (January, 2016).

[2] SAIL: Scalable and adaptive internet solutions. www.sail-project.eu, accessed on (January, 2016)

[3] Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., Ohlman, B, (2012) "A survey of informationcentric networking" Communications Magazine on IEEE 50(7), 26–36 .

[4] Alzahrani, B., Vassilakis, V., Reed, M, (2013) "Mitigating brute-force attacks on Bloom-filter based forwarding", Conference on Future Internet Communications (CFIC), pp. 1–7.

[5] Alzahrani, B., Vassilakis, V., Reed, M,(2014), "Selecting bloom-filter header lengths for secure information centric networking", 9th International Symposium on Communication Systems, Networks Digital Signal Processing (CSNDSP), pp. 628–633.

[6] Alzahrani, B., Reed, M., Riihijarvi, J., Vassilakis, V, (2015) "Scalability of information centric net-¨ working using mediated topology management", Journal of Network and Computer Applications, 50, 126-133.

[7] Alzahrani, B.A., Reed, M.J., Vassilakis, V.G, (2012), "Enabling z-filter updates for self-routing denialof-service resistant capabilities". Conference on Computer Science and Electronic Engineering (CEEC), 4th. pp. 100–105.

[8] Anderson, T., Roscoe, T., Wetherall, D, (2004), "Preventing internet denial-of-service with capabilities", SIGCOMM Comput. Commun. Rev. 34(1), 39–44.

[9] Antikainen, M., Aura, T., Sarela, M. (2014), "Denial-of-service attacks in Bloom-filter-based forwarding", IEEE/ACM Transactions on Networking, 22(5), 1463–1476.

[10] Bloom, B.H. (1970), "Space/time trade-offs in hash coding with allowable errors". Communications of the ACM 13, 422–426.

[11] Brunner, M., Abramowicz, H., Niebert, N., Correia, L.M. (2010), "4WARD: a european perspective towards the future internet". IEICE transactions on communications 93(3), 442–445.

[12] Carrea, L., Vernitski, A., Reed, M. (2014), "Optimized hash for network path encoding with minimized false positives". Computer Networks 58, 180–191.

[13] Choi, J., Han, J., Cho, E., Kwon, T., Choi, Y. (2011), "A survey on content-oriented networking for efficient content delivery". Comm. Mag. 49(3), 121–127.

[14] Dai, H., Wang, Y., Fan, J., Liu, B. (2013) "Mitigate ddos attacks in NDN by interest traceback", IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), pp. 381–386.

[15] Eugster, P.T., Felber, P.A., Guerraoui, R., Kermarrec, A.M. (2003), "The many faces of publish/subscribe". ACM Computing Surveys (CSUR) 35(2), 114–131.
[16] Feldmann, A. (2007), "Internet clean-slate design: what and why?" ACM SIGCOMM Computer Communication Review 37, 59–64.
[17] Ferguson, P., Senie, D. (1998), "Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing", rFC 2267.
[18] Fotiou, N., Polyzos, G., Trossen, D. (2011), "Illustrating a publish/subscribe internet architecture. Journal on Telecommunication Systems," Special Issue on Future Internet Sevices and Architecture: Trends and Visions 52, no.3.
[19] Fotiou, N., Nikander, P., Trossen, D., Polyzos, G.C. (2012), "Developing information networking further: From PSIRP to PURSUIT", Broadband Communications, Networks, and Systems, pp. 1–13. Springer.
[20] Frankel, S., Glenn, R., Kelly, S. (2003), "The AES-CBC cipher algorithm and its use with IPsec", RFC 3602
[21] Gajic, B., Riihijarvi, J., Mahonen, P. (2011), "Intra-domain topology manager for publish-subscribe networks", the 18th International Conference on Telecommunications (ICT), pp. 394–399. Ayia Napa, Cyprus.
[22] Gasti, P., Tsudik, G., Uzun, E., Zhang, L. (2013), "DoS and DDoS in named data networking", the 22nd International Conference on Computer Communications and Networks (ICCCN). pp. 1–7 .
[23] Handley, M. (2006), "Why the internet only just works". BT Technology Journal 24(3), 119–129.
[24] Interent live stats: "Internet users in the world", accessed on January (2006): http://www.internetlivestats.com/ internet-users
[25] Jacobson, V. (2006) , "A new way to look at networking". Google Tech Talks.
[26] Jacobson, V., Mosko, M., Smetters, D., Garcia-Luna-Aceves, J. (2007), "Content-centric networking". Whitepaper, Palo Alto Research Center pp. 2–4.
[27] Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M.F., Briggs, N.H., Braynard, R.L. (2009) "Networking named content". In: Proceedings of the 5th international conference on Emerging networking experiments and technologies. pp. 1–12. ACM, Rome, Italy.
[28] Jokela, P., Zahemszky, A., Esteve Rothenberg, C., Arianfar, S., Nikander, P. (2009), "LIPSIN: Line speed publish/subscribe inter-networking". ACM SIGCOMM Computer Communication Review 39(4), 195–206.
[29] Koponen, T., Chawla, M., Chun, B., Ermolinskiy, A., Kim, K., Shenker, S., Stoica, I. (2007), "A data oriented (and beyond) network architecture". SIGCOMM Comput. Commun. Rev. 34, no.4, 181–192.
[30] Kounavis, M.E., Kang, X., Grewal, K., Eszenyi, M., Gueron, S., Durham, D. (2010), "Encrypting the internet". SIGCOMM Comput. Commun. Rev. 41(4).
[31] Lagutin, D., Visala, K., Tarkoma, S. (2010), "Publish/Subscribe for internet: PSIRP perspective". Future Internet Assembly 84.
[32] Liu, X., Yang, X., Lu, Y. (2008), "To Filter or to authorize: Network-layer DoS Defense Against Multimillion-node Botnets". SIGCOMM Comput. Commun. Rev. 38(4), 195–206.
[33] Mathis, F.H. (1991), "A generalized birthday problem. SIAM Rev". 33, 265135–270
[34] Nikander, P., Marias, G.F. (2011), "Towards understanding pure publish/subscribe cryptographic protocols". In: Proceedings of the 16th International Conference on Security Protocols. pp. 144–155. Security'08, Springer-Verlag, Berlin, Heidelberg.
[35] Parno, B., Wendlandt, D., Shi, E., Perrig, A., Maggs, B., Hu, Y. (2007), "Portcullis: Protecting Connection Setup from Denial-of-capability Attacks". SIGCOMM Comput. Commun. Rev. 37(4), 289–300.
[36] Reed, M. (2012), "Traffic engineering for information-centric networks", IEEE International Conference on Communications (ICC). pp. 2660–2665. Ottawa, Canada.
[37] Rothenberg, C.E., Jokela, P., Nikander, P., Sarel¨ a, M., Ylitalo, J. (2009) "Self-routing Denial-of-Service¨ Resistant Capabilities using In-packet Bloom filters". Proceedings of the European Conference on Computer Network Defense (EC2ND). pp. 46–51. Milano Italy.
[38] Rothenberg, C.E., Macapuna, C., Magalh˜aes, M., Verdi, F., Wiesmaier, A. (2010), "In-packet Bloom filters: Design and networking applications". Elsevier Computer Networks 55, no. 6, 1364–1378.
[39] Sarel¨ a, M., Rothenberg, C.E., Aura, T., Zahemszky, A., Nikander, P., Ott, J. (2011), "Forwarding¨ anomalies in Bloom filter based multicast". In: 30th IEEE International Conference on Computer Communications (IEEE INFOCOM). Shanghai, China.

[40] Sarel¨ a, M., Esteve Rothenberg, C., Zahemszky, A., Nikander, P., Ott, J. (2012), "Bloomcasting: Security¨ in bloom filter based multicast". In: Proceedings of the 15th Nordic Conference on Information Security Technology for Applications. pp. 1–16. Springer-Verlag, Berlin, Heidelberg.

[41] Suzuki, K., Tonien, D., Kurosawa, K., Toyota, K. (2006), "Birthday paradox for multi-collisions". In: Proceedings of the 9th International Conference on Information Security and Cryptology. pp. 29–40. ICISC'06, Springer-Verlag, Berlin, Heidelberg.

[42] Tapolcai, J., Gulyas, A., Heszbergery, Z., Biro, J., Babarczi, P., Trossen, D. (2012), "Stateless multistage dissemination of information: Source routing revisited. In: IEEE Global Communications Conference (GLOBECOM). pp. 2797–2802. Anaheim, USA.

[43] Trossen, D., Parisis, G. (2012) "Designing and realizing an information-centric internet". IEEE Communications Magazine, 50(7), 60–67.

[44] Wendlandt, D., Andersen, D.G., Perrig, A. (2006), "Fastpass: Providing First-Packet Delivery". Tech. rep., CMU-CyLab.

[45] Yaar, A., Perrig, A., Song, D. (2004), "SIFF: a stateless Internet flow filter to mitigate DDoS flooding attacks". IEEE Symposium on Security and Privacy, pp. 130–143. Okland, USA.

[46] Yang, X., Wetherall, D., Anderson, T. (2008), "TVA: A DoS-limiting Network Architecture". IEEE/ACM Trans. Netw. 16(6), 1267–1280.

[47] Zahemszky, A., Csaszar, A., Nikkander, P., Rothenberg, C.E. (2009), "Exploring the pub/sub routing´ and forwarding space". IEEE ICC 09, Workshop on the Network of The Future. Dresden, Germany.