Efficient and secure lightweight Routing mechanism in Information Centric Network

Bander A. Alzahrani
Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

Abstract

Information-centric networking (ICN) architecture has been proposed to replace the current end-to-end Internet architecture to offer many advantages such as efficient routing, content caching and improved security. One of the promising proposed ICN model is the Publish-Subscribe Internet Technology (PURSUIT). One of the main characteristics that differentiate the PURSUIT architecture from the other ICN proposals is the use of Bloom filter, a space-efficient probabilistic data structure that offers simple efficient routing method with small forwarding tables. The PURSUIT model allows a number of forwarding mechanisms to be used, the Line Speed Publish/Subscribe Inter-networking (LIPSIN) and OptiHash are two candidates, which utilize the concept of Bloom filter. These mechanisms offer a highly efficient power consumption and high-speed routing, which makes it suitable for large-scale content distribution. However, they can be broken by denial of service attacks. In this paper, we study the security of these two routing mechanisms and propose a new secure forwarding mechanism that is able to resist DoS attacks with a very high probability, while maintaining the same advantages of existing Bloom filter-based solutions.

Keywords: Information Centric Networking, Bloom Filter, Security, Topology Management

Introduction

In the recent years, we are observing new communication paradigms coming into the scene and many established concepts are being revisited. One such new paradigm is the information-centric networking (ICN) (Xylomenos et al., 2014; Ahlgren et al., 2012) concept which has been attracting an increasing attention in the research community who attempts to apply its concepts into a wide range of communication networks, such as the Internet, mobile networks, and satellite networks. The appearance of ICN has been motivated by the intrinsic inefficiencies and security limitations of the current host-centric communication paradigm. Communication networks today are mainly used for delivering multimedia content to end users, rather than for connecting hosts. The user is interested in the content itself, while the location of the content is usually of minor importance or totally irrelevant. Also, security threats, such as distributed denial of service (DDoS) attacks can not be effectively mitigated under the host-centric paradigm.

Motivated by the aforementioned limitations of the current networks, the concept of ICN has emerged. ICN is a novel technology that provides access to named information objects as a native network service, aiming at improving the network efficiency and security. ICN aims at shifting the focus from the hosts to the information objects as well as to provide new and enhanced services to fixed and mobile users.

The PURSUIT architecture defines the following network entities: publishers, subscribers, and the mediation system. Publishers advertise the available information objects by issuing publication messages. Subscribers express their interest in receiving certain information objects by issuing

subscription messages. The mediation system comprises two functions: rendezvous (RV) and topology management (TM). These two functions control and manage the forwarding (FW) function. Each information object 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 the destination of publication and subscription messages sent by publishers and subscribers, respectively. When the RV detects a match (i.e., received publication and subscription messages for the same object), it contacts the TM who is responsible for constructing the delivery path/tree from the publishers to the subscribers. Once the path/tree has been established, the content can be delivered via a chain of FW nodes from the publisher to the subscriber using the LIPSIN mechanism. In practice this is done by encoding the delivery path into an in-packet Bloom filter that is used as the forwarding identifier (FId).

Herein, we are concerned with secure and efficient Bloom filter based content dissemination. One of the limitations of the LIPSIN mechanism is the probability of false positives. That is, it may happen that some packets are unnecessary forwarded via links that are not part of the delivery tree. This may increase traffic redundancy and cause security problems (Antikainen et al., 2014a). To mitigate the problem of false positives, a variant of the Bloom filter, called OptiHash, has been proposed (Carrea et al., 2014). However, OptiHash does not provide any security guarantees. For example, the attacker could perform several attacks such as flow duplication, packet aggregations, and forwarding loops, overwhelming the network resources (S¨arela¨ et al., April 2011). In this paper we a) apply the OptiHash approach for false positives reduction in Bloom filter-based forwarding, using as an example the LIPSIN mechanism in PURSUIT ICN and demonstrate the resultant improvements, and b) propose an enhancement of the OptiHash mechanism with security features. Comprehensive experimental results that have been conducted demonstrate appropriate levels of protection against DDoS attacks. The proposed approach also shows significant efficiency improvements over the traditional LIPSIN mechanism in terms of false positives reduction. This paper is organized as follows. In Section II, we present the related work on ICN security, focusing on DDoS attacks mitigation. In Section III, we briefly introduce the PURSUIT ICN architecture and the LIPSIN mechanism. In Section IV, we describe the OptiHah approach and its applicability to efficient Bloom filter based content dissemination in ICN. In Section V, we analyse the security issues of OptiHash and propose its security enhancement. In Section VI, we present our experimental results. We conclude in Section VII.

Related work

Here we present the related works on security of stateless forwarding in ICN. In (Rothenberg et al., 2009) the main security limitations of the LIPSIN mechanism have been identified. To address the security issues, the forwarding decision has been bound to packet contents and the processing context in a dynamic and computationally efficient manner. Hence, instead of using fixed FIds for each outgoing link, FIds are periodically computed and updated. This effectively means that different Bloom filters can be used to deliver content via the same path. This can improve the network security. Another security layer is added by demanding the FIds expire after a certain time and the subscriber will need to re-subscribe in case she wishes to continue receiving the content.

In (Alzahrani et al., 2013) the work of (Rothenberg et al., 2009) has been extended to prevent brute-force attacks on the LIPSIN mechanism. This has been achieved by calculating an optimal Bloom filter fill factor which results in a reduced DDoS attack probability. The fill factor refers to the number of bits set to 1 in the Bloom filter. In general, if the upper limit for the fill factor is set too high, then an attacker may encode a large number of link into the Bloom filter and, hence, the attack success probability will be high. On the other hand, if the fill factor is set too low, then the TM will not be able to encode large delivery paths/trees for legitimate subscribers. Hence, selecting an optimal fill factor can improve the network security while at the same time ensuring appropriate content delivery capability.

Another approach for DDoS mitigation in stateless Bloom filter based forwarding has been proposed in (Alzahrani et al., 2014b). The authors propose a method for optimally select the Bloom filter lengths. This results in an increased safe window and reduces the probability of a successful attack, since an upper bound for the fill factor can be further decreased.

In (Antikainen et al., 2014b) DDoS attacks in Bloom filter based forwarding are studied. A wide range of DDoS attacks is presented and the security limitations of the existing protocols are discussed. The

study indicates that the existing Bloom filter based mechanisms are not ready for deployment in real networks. Potential design options for secure and scalable Bloom-filter forwarding have been also investigated.

We observe that in the literature the following types security mechanisms have been considered: 1) limiting the number of links stored in the Bloom filters (Alzahrani et al., 2014b); 2) encrypting FIds so that they can not be reused or manipulated (Alzahrani et al., 2013); and 3) using cryptographically computed per-flow FIds (Rothenberg et al., 2009). In this work we adopt a centralized approach that optimizes the Bloom filter computation and, at the same time, prevents the attacker from manipulating the optimization parameters.

The PURSUIT ICN Model

The PURSUIT ICN model will serve as a basis for our security analysis (Alzahrani et al., 2014a). The publication/subscription process is as follows. The publisher notifies the RV that it can provide an information object. Similarly, the subscriber notifies the RV that it wishes to obtain an object. These two events need not be in the traditional client/server order. That is, the subscriber can request the object before the publisher has made it available. Each information object is given a statistically unique identifier, the RId. Each RId belongs to some scope that is used for object categorization. The scope is identified by the SId. Hence, the pair (RId, Sid) uniquely identifies an information object in the network. Once the publication and subscription messages are received by the RV, the latter is responsible for matching publishers and subscribers. When a match occurs, the RV contacts the TM and requests the construction of a delivery path/tree from the publisher to one or more subscribers. The TM is assumed to be aware of the network topology and encodes the path into the FId, which is a Bloom filter that contains the links of the constructed path. The FId is sent to the publisher. The latter inserts the FId into the packet header, appends the requested content, and sends the packet to its attached FW node. Finally, the FW nodes forward the packet along the encoded path until the packet reaches the subscriber. In Fig. 2 we present the basic Bloom filter based forwarding as realized by the LIPSIN mechanism. By using Bloom filters it is possible to encode a number of m-length link identifiers (LIds) into a single m-length FId. In particular, the forwarding operation is as follows. Each unidirectional link between two nodes is identified with an LId. This identifier is mbits long with k-bits set to 1, where k is the number of hash functions used to generate bit-positions set to 1. The number of bits set to 1 in an FId over the total number of bits is referred to as the fill factor. As previously mentioned, the selection of an upped bound for the fill factor may affect the efficiency and security of the network. As shown in Fig 1, the FId creation is done by performing the logical OR operation of the individual LIds constituting the tree.

Using the FId each FW node can decide where to forward the packets by performing a bitwise AND operation between the FId and the LIds of each outgoing link. If the result of this operation is true, then the FW node will forward the packet through the link assuming that this link is part of the delivery tree. It should be noted that, while the Bloom filters do not have false negatives, they may have false positives.

Optimising the filling factor by using the OptiHash

The OptiHash based forwarding: The OptiHash was originally proposed to control the false positive on the forwarding plane. It is considered a variant Bloom filter that aims to optimize the false positive rate at the stage of FID formation. It utilizes the same amount of space, 256-bit, that is used in the in-packet Bloom filter based forwarding approach (LIPSIN) proposed in (Jokela et al., 2009). It is a bit array w which consists of three parts: a bit array v of 241-bit long to store FID, the result of encoding the path, and two other slots of 7-bit and 8-bit long to store two values of integer parameters α and β (respectively). The parameters α and β are used to allow generating N = 27+8 alternative LIDs for each link.

The OptiHash creation: to illustrate the OptiHash forwarding approach, each link between two nodes is identified with two unidirectional links identifiers as in the LIPSIN forwarding approach (Jokela et al., 2009), but each link is 241-bit long instead of 256-bit. Initially, and at the network setup phase, the LIDs are created, and each forwarding node creates the LIDs for all its outgoing interfaces to be sent to the TM. To perform this, a set of k hash functions H = {h1,h2,...,h_k} is selected with the output

range in {1,2,..,241} (Knuth, 1998). Then for each link, the hash function H is applied with k hash functions. Another approach is to create the LIDs independently by the forwarding node and the TM so there is no need to send the LIDs by the forwarding nodes. The only information required to be known on both entities is the function H and the name of each link, so that each entity applies the hash over the names of links. Note that the values of the LIDs that are created at the setup phase are fixed.

When the TM receives a request from the RV, it determines the path between the publisher and subscriber/s and then forms the FID by having a bitwise-OR operation over all the fixed LIDs representing the path and checks for the presence of false positives. If a false positive is found, then the TM will also construct N new hashes for each LID on the path. The transformation from the fixed hash of LID to the new hash is performed by the following formula (Carrea et al., 2014):

$$\mu\alpha\beta = f\alpha\beta(\mu,\lambda) = (\mu + \mu\lambda\alpha + \lambda\beta) \bmod lv, \quad (1)$$

where $\mu = h(LID)$ is the current hash of the LID to transform, λ is the hash of the link where the packet was coming from, and α, β are the two parameters used for the transformation, and $\mu\alpha\beta$ is the transformed hash. $l(v)$ is the length of the array v to encode the path which is 241-bit.

Therefore, there will be 32768 candidate FIDs to be evaluated in terms of the false positives, so the best performing FID that offers a minimum false positives rate is selected. The pair (α, β) that corresponds to the FID with the minimum false positive is placed in the packet header along with the FID for forwarding. In the process of evaluating the FID and finding the optimal one, two strategies are used:

Optimising the false positive occurrence (the F-OptiHash): This optimisation is based on the actual number of false positive occurrences that can be found for practical sets of the LIDs. One set is E the one that represents the path (on-path links), and the other set is D which includes all the outgoing off-path links that are adjacent to the path and will be queried against the membership test. The LIDs in D are the links that may cause false positives. Since the false positives depend on the hashes of those links present in E and D, the set F becomes the set of links that will cause false positives and their hashes exist in both h(E) and h(D). To illustrate this more, consider the simple network graph in Figure 1, where the path from the publisher to the subscriber is represented by the links in set E = {11,5,14,17,13} and the set D = {16,4,66,7,2} is the links that will also be tested and may generate false positives. Assuming a hash function with k = 1 is used which means that the hash function is applied only once for each link to set one bit to 1. For simplicity we use the bit position of the resulted hash to represent the link instead of listing the whole 241-bit link identifier. Therefore, after hashing the set E and D the produced hashes are h(E) = {6,8,16,46,70} and h(D) = {3,6,9,24,6}. In this case a false positive occurs in the links F = {4,2} whose hashes are h(F) = {6,6}. Generally, the TM constructs the FID from the set E and takes a bitwise-AND operation with each LIDs in the set D and if the result is the same LID then a false positive occurs.

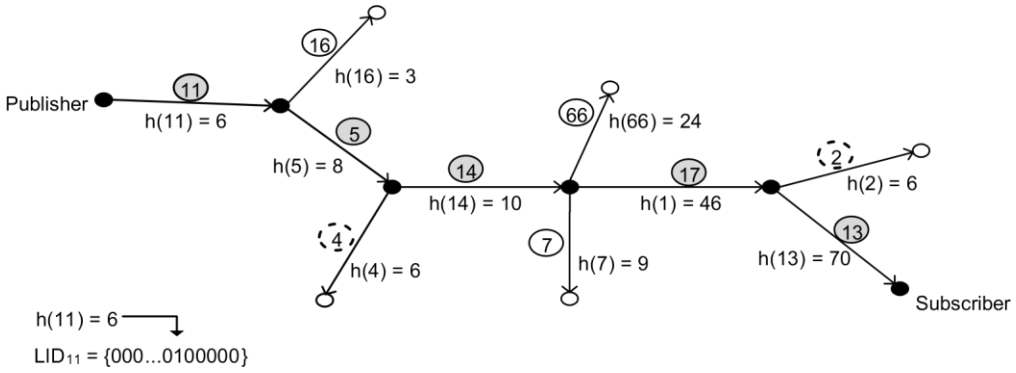

Figure 1: A simple network graph shows a path from a publisher to a subscriber.

The highlighted links are the set E that represent the path, whereas the others are the off-path links which represent the set D and the dotted links are the links that will generate false positives and form the set F. Each link in the graph is hashed with k=1.

Therefore, when the TM determines the path it initially calculates the hashes h(E) and h(D) and if a false positive is observed then instead of forming the FID with the fixed set of h(E), the TM will transform the LIDs using the function fαβ. For each pair of (α, β), the false positive rate is calculated by dividing the false positive occurrences |h(F)| by the number of links to be queried which may generate false positives |h(D)|:

$$f_p = \frac{|h(F)|}{|h(D)|} \tag{2}$$

Only the pair of (α, β) whose corresponding FID gives a minimum false positive rate is used for forwarding.

2. Optimising the filling factor (the ρ–OptiHash): since the false positive rate strictly depends on the filling factor of the FID, where an FID with a small value of ρ, will result in a lower rate of false positives. This obviously requires having lower amount of s (the number of 1s in the FID). Therefore, (Carrea, 2012) suggests that after finding the path between the publisher/subscriber(s), an optimisation on the filling factor is performed where the FID with the lowest filling factor is selected. To perform this, N = 27+8 FIDs are generated for the same path using all pairs of (α, β) and then for each FID the number of bits set to 1 (s) is counted. In this case, there will be N samples of FIDs to be evaluated in terms of the minimum value of s. Then the FID that contains the minimum number of 1s is selected along with its corresponding values of (α, β) to be placed in the packet header for forwarding.

The OptiHash forwarding decision: It is assumed that each forwarding node has the same function fαβ. Therefore, upon receiving a packet by the forwarding node it extracts the parameters α and β from the packet header to be used in recalculating the LIDs of its outgoing links by using the function fαβ. Then the FID is also loaded from the packet header and a membership test is performed by taking a bitwise-AND operation for each outgoing link with the FID. As in LIPSIN, if the result of the test gives the same LID then the packet is forwarded over that link assuming it is part of the path.

Security issues with the OptiHash scheme

There appears to be a security vulnerability that the OptiHash forwarding scheme does not protect from. We recall that the brute-force attack on LIPSIN forwarding scheme is launched by exploiting the false positives where the attacker sends random and maximally filled FIDs to the network without having any knowledge of the network's LIDs. The aim is to cause some false positives over the links until reaching the target. If the maximum allowed filling factor is high then the chance that the attacker gets to his victim in a short period increases. While the OptiHash is effective at reducing the false positive rate, it cannot be used in the context of mitigating bruteforce attacks. Since the forwarding nodes are unable to verify the received FID, whether it is the optimised one that was originally created by the TM along with its corresponding pair of (α, β) or not, the OptiHash mechanism can not be effective. Thus the attacker can still inject a random maximally filled FID to the network with any random values of α and β and in this case the forwarding nodes will accept and treat the packet as normal by using the injected parameters of (α, β) to recalculate the outgoing LIDs and then performing the membership test with the injected FID. In this case the attacker will cause the highest possible rate of the false positives as the FID used is maximally filled. Since the attack strategy does not require a pre-knowledge of the network LIDs, the probability of attack will still be same even with the new calculated LIDs that correspond to the random pair of {α, β}. Thus, the probability of the attack is the same as in the LIPSIN forwarding scheme.

The proposed secure OptiHash based forwarding

Since the ρ−OptiHash can support longer paths with smaller amount of filling factors compared with the classic LIPSIN forwarding scheme, one possible solution to mitigate brute-force attacks is to use the OptiHash forwarding scheme but with different optimisation aim. Since the probability of the attack depends on the amount of the filling factor used in the FID, the ρ−OptiHash optimisation can be proposed to restrict the attacker's ability of injecting an FID with a high value of the filling factor. In practice, we set an upper bound of maximum allowed filling factor in the network that gives an acceptable attack probability then for each request we optimse s, the number of 1s inserted in the FIDs and select the minimum. By this, we guarantee that the filling factor of any FID is always less than the upper bound limit and in the same time can support longer paths. Therefore, the attacker will not be able to create any FID with a number of 1s (s) higher than the maximum allowed limit selected for the network as it will be dropped. To perform this, initially two values are required be found at the network setup stage: the maximum number of LIDs that a unicast path could take (nm) in this network and the maximum number of 1s (s) that can be placed in any FID after the ρ−OptiHash optimisation for this nm. From this we find the optimised ρ_m by applying the following:

$$\rho_m = \frac{s}{m} \qquad (3)$$

where m is the length of the FID which is 241-bit.

Results and discussion

By using the concept of the OptiHash, we are able to minimise the filling factor and subsequently reduce the attack probability without affecting the network scalability of supporting large paths. The optimisation of s is shown in Figure 2 where the destitution of s for n = 25 LIDs is presented in both cases: the basic LIPSIN forwarding mechanism and the ρ-OptiHash. The left figure shows the maximum possible value of s which is 116 with ρ = 0.453 whereas we can optimise these values to s = 85 with ρ = 0.352 by using the ρ-OptiHash as it is shown in the right figure.

In (Alzahrani et al., 2013), we found that using ρm = 0.402, which supports a maximum path of nm = 23 links with 99.9% confidence, gives an acceptable safe window of 88.6 minutes. The safe window is the time required for an attacker to successfully guess a valid FID with a certain probability. However, in Figure 3 we show that the same path size (i.e. nm = 23) can be achieved with a smaller filling factor of 0.327. Therefore, with the ρOptiHash forwarding mechanism it becomes more difficult for the attacker to reach a target because the filling factor used is always optimised to the minimum and this reduces the probability of a successful attack.

As the ρ optimisation reduces the filling factor, the probability of attack is also reduced as shown in Figure 4. This figure shows the probability of launching a successful attack for a different number of LIDs to be encoded in the FID as a path for both forwarding mechanisms the (LIPSIN and the ρ-OptiHash). Overall, it can be observed that as n increases the optimisation of the ρ-OptiHash algorithm brings further improvements in the attack probability compared to that in the LIPSIN mechanism which is useful for the attack mitigation. For example, when using a path of n = 29 links, the probability of attack when LIPSIN is utilised is approximately $4 \times 10{-}8$ whereas by optimising ρ we can serve the same size of path and in the same time we can restrict the attack probability to approximately less than $7.8 \times 10{-}11$.

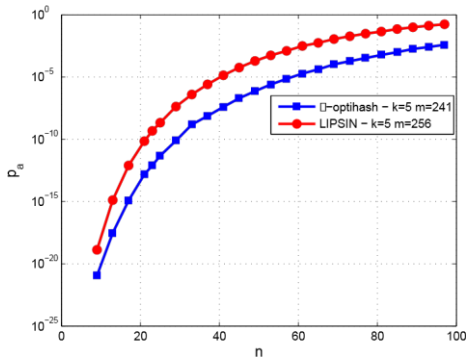

Figure 4: The probability of attack pa for different number of LIDs to be encoded into the FID using the ρ-OptiHash is compared with the probability of attack when the classic LIPSIN mechanism is used. The number of hash function used is k = 5 and the number of attack path length l = 5.

Now we turn to the improvement on the safe window when using the ρ-OptiHash instead of the LIPSIN forwarding mechanism. We demonstrate the safe window for the same attacking scenario used in (Rothenberg et al., 2009), where the node is able to send 106 packet/s with packet size 103 bits and 109 bit/s edge link to reach a specific victim 5-hop away with a successful attack confidence of pr = 0.5. The safe window is presented in Figure 5 as logarithmic scale to allow fitting the results into one plot. Overall, the results reveal a significant improvement on the safe window of the OptiHash over the LIPSIN as n increases and this is due to the increase of the hash collision explained earlier. However, as the optimisations in the safe window get better and better when n increases compared with those in LIPSIN. However, at some point these optmisations start to be less useful due to the short time in the safe window before and after the optimisation (couple of seconds). Since we are using in the parameters that give a longer safe window, the best results in all figures come when n does not exceeds 29 links.

Compared to the results presented in (Alzahrani et al., 2013) which suggests that for LIPSIN, the maximum path size should not exceeds nm = 23 links with k = 5 which gives a safe window of 14 × 102 seconds (less than 1 hour), with OptiHash we can support the same size of n but with a considerably longer safe window of 89.2×104 (247 hours). The network saleability may also be enhanced further to support longer maximum size of path up to nm = 29 and this can still give an acceptable safe window of approximately 2.5 hours when k = 5. In terms of the best value of k that can be used to offer a longer optimised safe window, Figures 11 to 14 show that k = 5 offers the best ρ-OptiHash performance among the parameters of k = {3,4,5,6} although k = {5,6} are similar.

Thus, if we use the ρ-OptiHash forwarding mechanism instead of the LIPSIN we need to frequently change the fixed LIDs that are created at the initialisation phase to prevent the attacker from achieving his in-progress attack even if the probability is very low because after certain attempts this probability will be achieved. To change the LIDs, just like the zFormation technique proposed in (Rothenberg et al., 2009), the TM needs to have a synchronised clock and shares a time bound secret key with each forwarding node in order to change the LIDs independently. This means that every (Δt), the TM needs to update all in-progress FIDs as they about to expire. In fact, the expired FIDs may still be accepted by the forwarding node and falsely forwarded, however the advantage of the optimisation will be lost as the main fixed set of LIDs will have been changed.

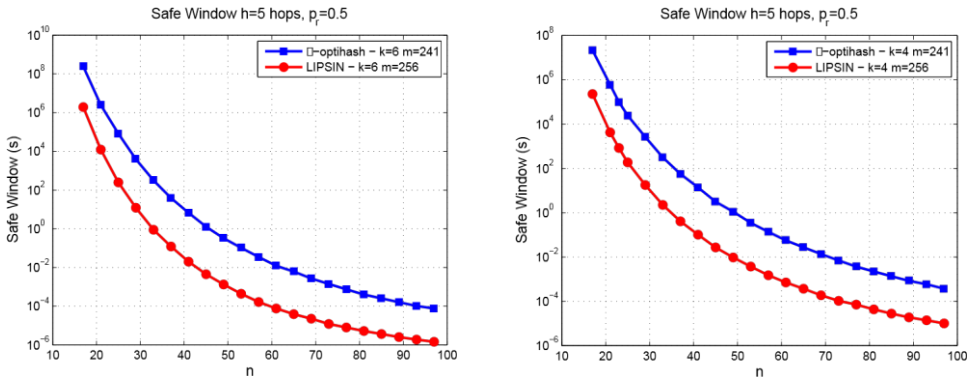

Figure 4: The improvement in the safe window for different number of LIDs to be encoded into the FID when using the ρOptiHash is compared with the safe window of the classic LIPSIN mechanism. The number of hash function used is k = 6 (left), 4 (right), the number of attack path length is h = 5 with probability of successes pr = 0.5.

If we setup the network with k = 5 to support a maximum path of nm = 23 which gives a maximum filling factor of ρm = 0.327 links and choose (Δt) = 40 minutes to change the fixed LIDs, the probability pr that the attacker successfully reach a victim that is 5-hop away within this (Δt) is calculated as follows:

$$x = 40 \times 60 \times 106 = 2.4 \times 109 \text{attempts.} \quad (4)$$

$$pa = 0.3275^{\times 5} \approx 7.3 \times 10{-}13. \quad (5)$$

$$pr = 1 - (1 - pa)x \approx 0.0017. \quad (6)$$

8

This probability pr sinks drastically compared to the probability pr ≈ 0.26 found when LIPSIN is used with the same attack scenario and the same size of path nm = 23 links. This difference in the probability is considered a significant improvement towards securing the network. Moreover, this improvement is to support a maximum path of n_m = 23 and in case of severing a multicast request that forms a path of n ≥ 23 links or generally greater than the specified nm then the TM divides the requested path into smaller ones each with n ≤ n_m.

It should be noted that the advantage of using ρ-OptiHash hash to mitigate brute-force attacks comes at the cost of having higher processing time and an increased overhead at the TM. To find the time complexity of the ρ-OptiHash at the TM when creating the optimal FID in terms of filling factor, the algorithm performs the following steps and for each step the time complexity is stated (Carrea, 2012). (1) The TM has first to calculate the new hashes μαβ for each pair of (α,β) of each element in the path E using (1) with k hash functions, then it has to find the corresponding filling factor for the N candidates FIDs (N = 27+8). The time complexity of this is O(Nk|E|). (2) Then the TM has to select the pair of (α,β) that offers the best performing FID which has minimum filling factor, and the time complexity of this is O(N). Therefore, the overall complexity of the algorithm is O(N + Nk|E|) which can be approximated as O(Nk|E|).

Conclusion

Information centric network has been proposed to tackle many issues exist in the current host-centric Internet network architecture. There are a number of ICN models that are being proposed with

differing models for contents dissemination and delivery. One of the key goals that these ICN models share is to consider eliminating the DDoS attacks issue in the original design, by insuring equal balance between publisher and subscriber and eliminates the power that senders have in the current IP network. Therefore, the security of these ICN solutions needs careful inspection if we are to move from early prototypes towards proposing them as models for a future network operating at an open environment. In this paper we have considered an ICN architecture following the architecture proposed by the PURSUIT which utilise the concept of Bloom filter-based delivery in the forwarding plane. Since all existing forwarding based Bloom filters approaches result in a low rate of false positives, which may be exploited by malicious users to inject any arbitrary traffic. We have studied the security of existing forwarding mechanisms, and propose a new approach that provide mitigation of DDoS. We have shown that our solution can restrict the attacker capability, while supporting larger delivery tree.

References

Ahlgren, B., Dannewitz, C., Imbrenda, C., Kutscher, D., Ohlman, B., 2012. A survey of information-centric networking. Communications Magazine, IEEE 50 (7), 26–36.

AL-Naday, M. F., Thomos, N., Reed, M. J., 2016. Information-centric multilayer networking: Improving performance through an icn/wdm architecture.

Alzahrani, B., Reed, M., Riihij¨arvi, J., Vassilakis, V., July 2014a. Scalability of information centric networking using mediated topology management. Journal of Networkand Computer Applications, ElsevierWww.dx.doi.org/10.1016/j.jnca.2014.07.002.

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

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

Antikainen, M., Aura, T., Sa¨rela¨, M., 2014a. Denial-of-service attacks in bloom-filter-based forwarding.

IEEE/ACM Transactions on Networking (TON) 22 (5), 1463–1476.

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

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

Carrea, L., 2012. Optimised probabilistic data structures for forwarding in information centric networking. Ph.D. thesis, Univeristy od Essex, UK.

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

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

Edwall, T., 2011. Scalable & adaptive internet solutions (sail).

Fotiou, N., Nikander, P., Trossen, D., Polyzos, G. C., 2012. Developing information networking further: From PSIRP to PURSUIT. In: Broadband Communications, Networks, and Systems. Springer, pp. 1–13.

Gasti, P., Tsudik, G., Uzun, E., Zhang, L., 2013. DoS and DDoS in named data networking. In: Computer Communications and Networks (ICCCN), the 22nd International Conference on. pp. 1–7.

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. ACM, Rome, Italy, pp. 1–12.

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.

Knuth, D., 1998. The Art of Computer Programming - Part 3: Sorting and Searching. Wesley.

Koponen, T., Chawla, M., Chun, B.-G., Ermolinskiy, A., Kim, K. H., Shenker, S., Stoica, I., 2007. A data-oriented (and beyond) network architecture. ACM SIGCOMM Computer Communication Review 37 (4), 181–192.

Mamatas, L., Papadopoulou, A., Tsaoussidis, V., 2015. Exploiting communication opportunities in disrupted network environments. In: International Conference on Wired/Wireless Internet Communication. Springer, pp. 180–193.

Pavlou, G., Wang, N., Chai, W. K., Psaras, I., 2013. Internet-scale content mediation in information-centric networks. annals of telecommunicationsannales des t´el´ecommunications 68 (3-4), 167–177.

Rothenberg, C. E., Jokela, P., Nikander, P., Sa¨rel¨a, M., Ylitalo, J., 2009. Selfrouting Denial-of-Service Resistant Capabilities using In-packet Bloom filters. In: Proceedings of the European Conference on Computer Network Defense (EC2ND). Milano Italy, pp. 46–51.

Sa¨rel¨a, M., Rothenberg, C. E., Aura, T., Zahemszky, A., Nikander, P., Ott, J., April 2011. Forwarding anomalies in Bloom filter based multicast. In: 30th IEEE International Conference on Computer Communications (IEEE INFOCOM 2011). Shanghai, China.

Vassilakis, V., Moscholios, I. D., Alohali, B. A., Logothetis, M. D., 2015. Mitigating distributed denial-of-service attacks in named data networking. Proc. AICT, 21–26.

Vassilakis, V. G., Wang, L., Carrea, L., Moscholios, I. D., Logothetis, M. D., 2016. Scalable bloom-filter based content dissemination in community networks using information centric principles.

Wang, L., Bayhan, S., Ott, J., Kangasharju, J., Sathiaseelan, A., Crowcroft, J., 2015. Pro-diluvian: Understanding scoped-flooding for content discovery in information-centric networking. In: Proceedings of the 2nd International Conference on Information-Centric Networking. ACM, pp. 9–18.

Xylomenos, G., Ververidis, C., Siris, V., Fotiou, N., Tsilopoulos, C., Vasilakos, X., Katsaros, K., Polyzos, G., Second 2014. A survey of informationcentric networking research. Communications Surveys Tutorials, IEEE 16 (2), 1024–1049.