Roles and Security in a Publish/Subscribe Network Architecture

Dmitrij Lagutin, Kari Visala, Andras Zahemszky
Helsinki Institute for Information Technology HIIT
Helsinki University of Technology TKK
Espoo, Finland
Email: firstname.lastname@hiit.fi

Trevor Burbridge
BT Research
British Telecom
Ipswich, UK
trevor.burbridge@bt.com

Giannis F. Marias
Mobile Multimedia Lab
Athens University of Economics and Business
Athens, Greece
marias@aueb.gr

***Abstract*—Several *publish/subscribe* (pub/sub) and data-oriented networking proposals have been presented to overcome limitations of the current message- and host-centric Internet. However, security issues of these solutions have not been addressed comprehensively. In this paper we examine roles of actors comprising an inter-domain pub/sub network, together with security requirements and minimal required trust associations arising from this setting. We then introduce and analyze a security design for a clean-slate pub/sub network architecture that secures both the control and data planes. The solution addresses availability and data integrity while remaining scalable and usable.**

***Keywords*-Publish/subscribe networking; network security; denial-of-service; future network architectures.**

I. Introduction

Traditionally the Internet has been a host- and message-centric system, which has lead to several problems in terms of security, scalability and mobility. Since the sender is in complete control of communication, denial of service attacks are easy to launch. Additionally, an efficient multicast is difficult to implement on the Internet's scale and since the IP address acts as both the node identifier and locator, mobility is problematic to achieve. To overcome these problems, a data-oriented *publish/subscribe* (pub/sub) networking approach has been proposed [1], [2], [3]. Instead of making connections between hosts, the data is published and subscribed to, and the network makes the best effort to deliver data objects to subscribers regardless of their location. Several kinds of applications would greatly benefit from such approach, for example, simultaneous video streaming to a large amount of users, or novel collaboration applications like Google Wave. However, most of existing pub/sub proposals are overlay solutions based on IP, suffering from the same underlying problems as the current Internet. The security aspects of pub/sub networking have also been mostly ignored.

This paper describes a secure, network layer clean-slate data-oriented pub/sub internetworking solution that does not use IP at all. The contributions of this paper include description of security roles in the pub/sub architecture, the security architecture that provides availability for all aspects of pub/sub networking, including the rendezvous and forwarding, and the analysis of the architecture.

The paper is organized as follows. The basic concept of pub/sub and our goals are described in Section II. Our pub/sub architecture is explained in more detail in Section III. Section IV describes our security solution, which is analyzed in Section V. The related work is discussed in Section VI while Section VII concludes the paper.

II. Basic Concepts and Requirements

In the abstract data-oriented publish/subscribe model the communication between publishers and subscribers is decoupled in time and space by the publication in the middle. Each publication is identified by an identifier that is persistently associated with the data content of the publication.

On the most basic level, our data-oriented pub/sub network architecture can be viewed as having four distinct parts: publisher-side entity hosting the data, subscribers, the rendezvous system and the routing/forwarding planes spanning over the inter-domain topology along which payload data is delivered. The publisher-side entity advertises potential publications in the rendezvous system and serves the data contents to the forwarding layer when it receives a new subscription via the routing layer. From the security point of view, the rendezvous system is the most crucial component of the system. It acts as a middleman between publishers and subscribers, and is involved in configuring the forwarding path for data delivery.

A. Problem Description

The main security goals for our network layer pub/sub architecture solution are outlined below. Confidentiality and privacy fall outside the scope of this paper as they can be addressed on higher layers. Otherwise, our goals are similar to ones presented in [4].

Preventing unwanted traffic and availability. In order to protect the network from denial of service attacks and its users from SPAM, unwanted traffic should be stopped by the network. This requirement applies both to the rendezvous traffic, i.e., subscribers cannot flood publishers and rendezvous nodes with bogus subscription messages, and for the forwarding layer meaning that no data should be delivered unless there is a valid subscription from the subscriber.

978-1-4244-7755-5/10/$26.00 ©2010 IEEE

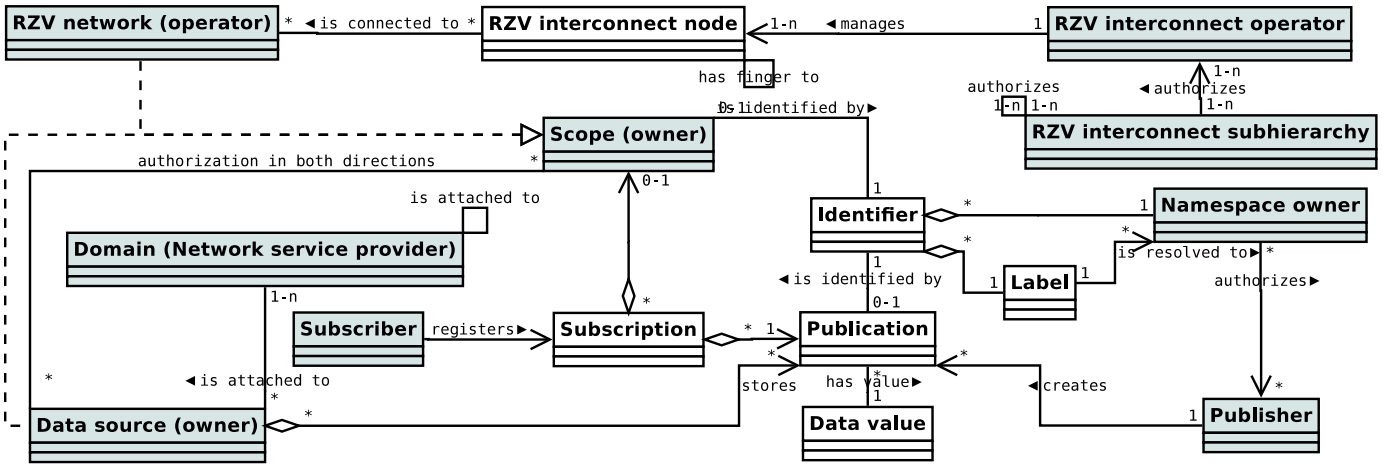

Figure 1. Roles of actors in the system, main concepts, and some of their relationships are depicted as a static diagram. Roles are marked with a darker background color. We have combined some entities and their owners into a single box to save space in the diagram.

Prevention of unwanted traffic will improve availability, since all parties will be able to serve valid users.

Integrity. Subscribers should be able to verify that the content of the received publication is not compromised.

Scalability. The system should be scalable to Internet core networks and wireless, low power devices.

Since the Internet is a huge network composed of thousands of networks and millions users, we cannot assume that all nodes in the network will be benevolent. Hence, above mentioned requirements should be satisfied even if some core parts of the network are hostile.

III. Roles and Architecture of Clean Slate Publish/Subscribe Networking

Based on the general pub/sub model described in Section II, we now introduce a more fine-grained model. A more detailed description of the architecture is available in [1].

The main actors of the system are listed below while relationships between actors are shown in Figure 1. The same actor can have multiple roles, often the namespace owner, scope owner, publisher, and data source are one entity.

Namespace owners manage namespaces for publication identifiers. The namespace owner authorizes publishers to use part of the namespace for their publications.

Publishers create the actual publications, which are in turn delivered to the *Subscribers*.

Data sources at the edge of the network store the publication contents persistently and serve it to the subscribers.

Rendezvous networks (RN) provide the rendezvous service to scopes, data sources and subscribers.

Scopes are abstract entities that control how publications are disseminated. The scope authorizes one or more data sources to host the publication data, and RNs to store its advertisements. Such RN is called *home rendezvous network* (HRN). The scope also functions as the policy decision point for the access control for scope contents for subscribers.

Rendezvous interconnect operator. Rendezvous networks are interconnected using a hierarchical Chord DHT [5] implementation, which is responsible for storing global scope advertisements on behalf of the originating RNs. When a subscriber initiates a rendezvous and the publication cannot be found from the local rendezvous network, the rendezvous request is recursively routed using the Crescendo algorithm with the modification that each message is actually a subscription operation on top of the routing layer explained in [1]. The results of the rendezvous operations can be cached in RNs and the interconnect nodes. *Rendezvous interconnect operators* are the organizations that provide the nodes for the interconnect architecture.

The *rendezvous interconnect subhierarchy owner* controls each subhierarchy of the RI. Together these owners authorize RI nodes to join the part of the hierarchy of the overlay and provide them with an address range from the Chord ring.

A. Identifiers and Rendezvous

On the network layer publications are identified by rendezvous identifiers (Rid), while scope identifiers (Sid) denote scopes. To access the data the subscriber must know both the Rid and Sid[1]. Rids and Sids have a P:L (public key:label) structure similar to DONA [3]. We use elliptic curve cryptography (ECC) [6] for cryptographic keys and signatures. Since a 163-bit EEC key offers the same cryptographic strength as a 1024-bit RSA key [7], we can include a whole public key in the P part of the Rid. In the payload the label part of Rid contains a hash of the arbitrary label, while the rendezvous requests and subscription messages can include the original variable length label to enable dynamically generated content.

The rendezvous process is depicted in Figure 2 and works as follows. The data source sends first a publish request

[1]A separate mechanism is needed to resolve human readable long-term names to <Rid:Sid> pairs, but it is outside the scope of this paper.

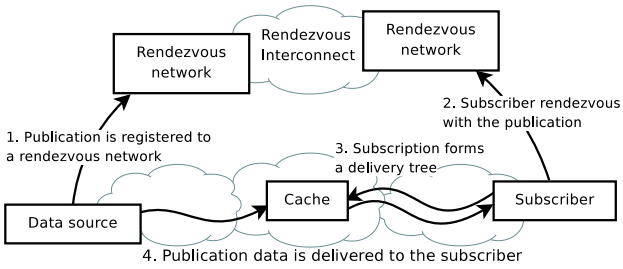

Figure 2. In order to be scalable while using efficient payload routes, the rendezvous is separated as a modular layer from routing and forwarding.

to the rendezvous. The subscriber that is interested in data sends first a rendezvous request, and receives the topological location of the data in a reply from the rendezvous system. Finally, the subscriber sends a subscription request towards the data source. If any node on the path has already the data in its cache, the cached data will be sent to the subscriber.

B. Forwarding

As forwarding solely based on Rids would imply poor scalability properties (state explosion) for the forwarding architecture, the system requires a separate forwarding plane that acts on forwarding identifiers (Fids). The forwarding architecture inherently supports multicast, while unicast is treated as the degenerate case of multicast with a single receiver. Furthermore, there are strict security requirements for the forwarding structure, as it should have built-in DDoS-resistance. Basically, the Fid should act as capability and the forwarding plane should deny forwarding publications without valid subscriptions. In our system, the Fid is a *Bloom filter* [8] that encodes the whole or part of the forwarding path/tree the publication should traverse.

IV. Security Mechanisms

This section describes our security solution for the publish/subscribe networking. Our goal is to allow system's particants to independently verify whether other participant are adhering to rules. Therefore, misuse can be easily noticed providing an incentive for everyone to adhere to the rules.

Main security components include packet level authentication (PLA) that is used to secure rendezvous and subscription control plane messages, rendezvous authorization through certificates, secure rendezvous interconnect routing, and zFormation that protects the forwarding plane.

A. Packet Level Authentication (PLA)

Packet Level Authentication (PLA) [9] is a novel way to provide availability, accountability and to protect the network infrastructure from several kinds of attacks, like denial-of-service (DoS) attacks, on the network layer. PLA is based on the assumption that per packet public key cryptographic operations are possible at wire speed in high speed networks due to new cryptographic algorithms and advances in semiconductor technology.

PLA aims to detect and stop malicious traffic as quickly as possible freeing resources to the benevolent traffic. A good analogy to the principle of PLA is a paper currency. Anyone can independently verify whether the bill is authentic simply by checking the security measures inside the bill like a watermark and hologram. There is no need to contact the bank that has issued the bill. Similarly, PLA gives every node a possibility to check whether the packet has been modified, duplicated or delayed without a previously established trust relation with the sender of the packet. Such packets can discarded immediately by any node in the network.

PLA adds an own header to the packet, which among other things, contains the sender's public key and the signature over the packet. This signature protects the integrity of packet and offers non-reputability. Since PLA includes signatures and public keys in every packet, it is not feasible to use traditional cryptographic solutions like RSA, therefore PLA uses elliptic curve cryptography (ECC) [6]. An FPGA-based hardware accelerator has been developed for PLA [10] to accelerating cryptographic operations.

PLA is used to secure rendezvous and subscription messages in various situations described below, and can optionally be used to secure the payload traffic.

B. Certificates and Network Attachment

Our security solution for rendezvous is mostly based on traditional certificates consisting of issuer, subject, rights, and validity time fields. In most cases validity time is short to reduce the window of vulnerability and eliminate the need for a centralized revocation system.

When a node, for example a data source or subscriber, is bootstrapped, it will receive a certificate from the local access network. This certificate acts a proof that the node has permission to use the network and it also provides accountability. For privacy reasons, nodes may posses multiple cryptographic identities and associated certificates. In the following examples CX denotes the certificate from the access network to the subscriber and CY denotes a similar certificate to the data source. The certificates are expressed in the S-Expression [11] format.

C. Rendezvous Authorization

Figure 3 describes the whole publish/subscribe related signaling. In the step 0, the scope authorizes the data source to serve the publication (<Sid:Rid>) by C1 certificate, while the data source acknowledges its willingless by C2 certificate. Without the C2 certificate, a hostile scope could induce load to the data source by claiming that the publication can be found from the target.

In the first step, the data source sends a publication advertisement concerning <Sid:Rid> to the rendezvous system. This message contains the data source's certificate from the access network (CY), along with C1 and C2 certificates. The whole message is protected by data source's signature. In the

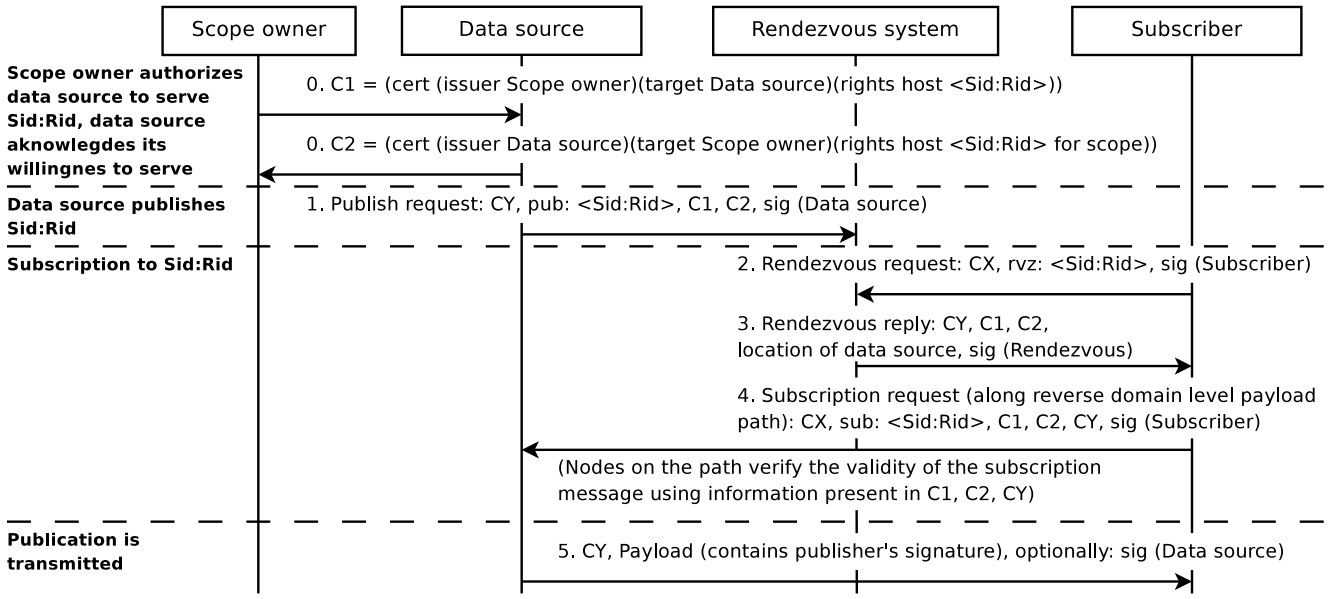

Figure 3. The phases of communication: 0. scope configuration, 1. advertisement of publication, 2-4. subscription sets up a delivery tree, 5. payload transfer. The sequence shown is a slight simplification of the actual system as, for example, caches are not included in the diagram for clarity.

next steps, the subscriber initiates the subscription process and the rendezvous system returns a data source's location within the network along with all relevant certificates.

The final subscription message is sent in the step 4. This message contains CX, CY, C1 and C2 certificates, which together with the P:L structure of identifiers offer proof to intermediate nodes that this message is valid. Any node on the path can verify that the scope and data source have authorized each other to serve <Sid:Rid>, and the data source is really a valid entity in the network.

Finally, in step 5, the publication is delivered from the data source to the subscriber. The payload delivery may be optionally protected by data source's signature using PLA.

1) Access Control for Subscriptions: While the example above presents a basic case of pub/sub flow, our system also supports more advanced optional features. For some use cases it is important to have an access control for subscribers for a certain scope. Ultimately, the labels and content of the private publications must be encrypted to prevent network caches from leaking the content, but the routing layer also prevents unauthorized subscription messages from reaching the data source. If it is also important to keep the location of data sources confidential, additional checks can be implemented in the rendezvous system.

The example presented in Figure 3 can be augmented to support the access control. In this case the C1 certificate would contain different rights, meaning that access control is required and therefore a subscriber must posses a separate certificate (C3) from the scope, which acts like a capability. In step 2 the rendezvous request will go all the way to the scope, and subscriber must authenticate itself, e.g., using a password or other means. Such authentication may require additional signaling that is not shown in the example. After a successful authentication, the subscriber will receive the above-mentioned C3 certificate from the scope, and will include it to subscription message in step 4. We note that the C3 capability can also be disseminated by the higher layers.

In step 4 intermediate nodes will see from the C1 certificate that the access control check is present, and will verify the subscription message will contain a valid C3 certificate. The issuer field of C3 certificate must match with the public key present in Sid, while the subject field must match with subject field of the CX certificate. This way a hostile subscriber can not reuse a valid C3 certificate.

D. Rendezvous Interconnect Security

The availability of the rendezvous service is secured with the help of RI subhierarchy owners that act as a trusted third parties authenticating RI nodes before they can join the DHT [12]. Each RI node is assigned an identifier range from the Chord ring using a temporary certificate signed by the RI subhierarchy owner in question. This prevents the *Sybil attack* [13] and the routing table poisoning attack and we can assume that only a relatively small portion of the nodes is malicious, which makes replication an effective solution to availability. The locality optimized links in the DHT are based on the Canon hierarchy that is expected to loosely follow the underlying network topology.

Scope authorizes a RN to host itself with a certificate, which prevents false scope advertisements in the RI. The Canon routing algorithm guarantees that the most local advertisement in the RI hierarchy is always found first. To avoid DDoS attacks against particular RI nodes, we utilize traffic *admission control* and *forwarding limit* at each node of the RI as presented in [14, p. 134]. This is possible,

because the Sids are uniformly hashed to the DHT nodes and honest traffic distribution should be roughly balanced when taken caching into account. As a result, the same subscriber may not continuously rendezvous with the Rids of the same scope and has the incentive to cache the results of the rendezvous operation. RI nodes also cache popular rendezvous results and store a subscription in the RI to monitor updates of the cached data. This makes popular data scalable without large investment in the HRNs by the scope owner. The RI is further protected from attacks by the fact that it uses the pub/sub model for communication provided by the underlying routing and forwarding layers which makes it difficult to circumvent the DHT topology. Access controlled scopes require the rendezvous message to reach a HRN trusted by the scope that can act as a policy enforcement point and encrypt the response with the public key of the subscriber. Popular access controlled scopes require the rendezvous network a large capacity for handling incoming subscriptions and possible DDoS subscription attacks by botnets. To address this, it is possible to replicate the scope implementation to multiple RNs. To avoid a RI node becoming a hotspot, it is possible to create multiple advertisements in the RI by adding salt values [15] to the Sid before hashing it to determine the RI node.

On the rendezvous system level we do not have to consider the tussle for good human-readable names as the scope names are always relative to a namespace created by the public key of the Sid. On the other hand, each scope advertisement consumes storage resources of the RI and a fair allocation should be enforced by the RI mechanism. This can be achieved, for example, with per-node quotas for each subhierarchy in the RI, which gives each subhierarchy the incentives to control resource usage of each of its clients. The quotas can be based on contracts between interconnect operators and their customers.

E. Forwarding Using zFormation

For forwarding a form of source routing is used instead of hop-by-hop routing decisions. Instead of naming nodes, our architecture names directional links between the nodes. The Fid is a small Bloom filter, called zFilter [16] that encodes all the link identifiers (LIDs) the packet should traverse. The forwarding decision is a simple membership check, where the forwarding node checks which of its outgoing LIDs are present in the zFilter and forwards accordingly. As Bloom filters are probabilistic data structures, false positives may happen, when the zFilters contain too many LIDs. However, this basic mechanism with 256-bit Bloom filters is capable of handling around 40 unicast or multicast hops with acceptable bandwidth efficiency. Mechanisms for supporting larger trees include introducing state by switching the zFilters in dedicated nodes.

If an attacker manages to learn valid zFilters, it will be able to send traffic on particular trees and the forwarding nodes will forward the unwanted traffic towards the uninterested receivers. Therefore, we use the zFormation [17] mechanism, which provides a complete forwarding plane security. First, zFilters are valid only for a certain amount of time. Second, the zFilter is tied to the forwarding tree, making it impossible to reuse the zFilter from locations other than the original source, preventing e.g. injecting traffic into the middle of the tree or combining zFilters together. Third, the zFilter is tied to the Rid of the publication. As a consequence, the attacker cannot use an otherwise valid zFilter with different publications, only for those it got an explicit permission by the rendezvous system.

A special function, which can be a streamcipher-like function, is used to implement zFormation. Instead of the membership check described above, now the forwarding node computes the output value of this function and compares it with the zFilter. More specifically, for each outgoing link, the special function will produce a LID based on a periodically changing secret key K(t), on the incoming interface, and the Rid found in the packet header. The state requirement is even lower than with the basic zFilters, though the price is the increased computational complexity. To be able to compute valid zFilters, the entity responsible for this task needs to know the secret keys of the of the individual forwarding nodes, in addition to the topology knowledge. In other words, the zFilter is computed by applying the same function that is used for forwarding decisions.

V. Analysis

Our solution protects the rendezvous signaling traffic by PLA and cryptographic signatures, and additionally the data identifiers (Rids and Sids) are tied to cryptographic identifiers. This creates a strong binding between the actual data and the network traffic, effectively preventing or mitigating most of attacks against the system. Finally, zFilters prevent denial of service attacks on the forwarding layer.

Separation of the namespace owner, publisher, and data source removes a single point of failure and adds more flexibility and security in the system. For example, if the data source has been compromised, its certificate will not be renewed by the scope and the same Rid can be served from a different data source.

Several related security solutions, such as AIP [18] or DONA [3], utilize hashes of cryptographic keys as identifiers. Since our solution uses compact ECC keys, the whole public key can be included in Rids and Sids. This reduces the overall bandwidth overhead and simplifies the security solution, because the public key does not need to be separately included in the payload.

Below we analyze how our solution satisfies the requirements presented in Section II.

Prevention of unwanted rendezvous traffic and availability. In our architecture, the most important issue is to prevent nodes from flooding subscription messages towards

data sources, since these messages are transmitted globally. Rendezvous messages are transmitted through the local rendezvous service, which can be easily load balanced and therefore there is no risk of DDoS attack.

Our solution allows nodes on the path to independently verify validity of rendezvous and subscription messages, and distinguish those messages from the data traffic. In this case the validity means that subscriber and data source are valid entities in the network, subscriber wants to receive the publication, the data source is willing to serve the publication and is authorized by the scope. Independently verifiable per-subscriber capabilities are also supported.

Cryptographic identities and signatures allow the network to differentiate users and traffic easily. Therefore the access network can limit amount of subscription messages sent by a single node within a given timeframe, and subscription messages can also be limited per destination. Since the size of the subscription message with all certificates is few kilobits at most, this effectively prevents severe DDoS attacks. Subscription messages can also be prioritized higher than the data traffic. Therefore in a case of congestion, subscriptions will get through at the expense of bulk data transfer, improving the overall availability of the network.

The network attachment mechanism allows removal of malicious users from the network altogether. The certificate given by the access network may be valid only for short periods of time, and it will not be renewed if the user engages in malicious behavior.

Independent verification of subscription messages also provides accountability, since the next domain on the path can check whether the previous domain has dropped invalid messages. This gives an incentive for all parties to obey the rules, and drop invalid messages as soon as possible.

Preventing unwanted payload traffic. zFilters together with zFormation mechanism provide a good protection against unwanted traffic. Since the zFilters are not globally known, it is impossible for the attacker to send data to a specific target. Even if some information about zFilters is leaked, zFormation method of limiting the zFilter to a some specific forwarding tree and Rid prevents DDoS attacks. For individual popular publications a subscription-based DDoS attack is not a problem as the network implicitly forms a multicast delivery tree consuming mostly subscribers' resources.

Integrity of publications is protected by publisher's signature, therefore subscribers can verify that the publication data has not been tampered with.

Scalability. By default, we use per-packet signatures and certificates only for few control messages during the rendezvous phase, therefore the computational overhead is not significant. The per-packet bandwidth overhead is also relatively low, less than 500 bits for payload traffic compared to the plain IPv6.

The rendezvous system itself is designed to be scalable. For example, rendezvous nodes and more capable routers can cache rendezvous reply messages. Therefore if there are multiple subscribers accessing public data, there is no need for every subscriber to contact the original rendezvous node.

The zFilter forwarding is fast and scalable since it is mostly based on simple "AND" operations. We have implemented zFilter using the NetFPGA [19] platform, and the performance results are promising [16].

VI. Related Work

This section covers related work for publish/subscribe systems and network layer security solutions.

A. Data-oriented and Publish/subscribe Systems

A data-oriented network architecture (DONA) [3] replaces a traditional DNS-based namespace with self-certifying flat labels, which are derived from cryptographic public keys. DONA names are expressed as a P:L pair, where P is a hash of a principal's public key which owns the data and L is a label. DONA utilizes an IP header extension mechanism to add a DONA header to the IP header, and separate resolution handlers (RHs) are used to resolve P:L pairs in topological locations.

In content-centric networking (CCN) [2] every packet has an unique human-readable name. CCN uses two types of packets. Consumers of data send interest packets to the network, and a nodes possessing the data reply with the corresponding data packet. Since packets are independently named, a separate interest packet must be sent for each required data packet. In CCN data packets are signed by the original publisher allowing independent verification, however interest packet's are not always protected by signatures.

Security issues of the content-based pub/sub system have been explored in [20]. The work proposes secure event types, where the publication's user friendly name is tied to the publisher's cryptographic key.

B. Security Mechanisms

Most of existing network layer security proposals utilize hash chains or Merkle trees [21]. Examples of hash chain based solutions include TESLA [22], which is time-based hash chain scheme, and ALPHA [23] that relies on interaction between the sender and receiver. While hash chain approaches are very lightweight, they have several downsides, such as path dependency and complex signaling. Merkle tree based solutions have high bandwidth overhead for large trees, and their performance suffers if packets arrive in out-of-order.

Accountable Internet Protocol (AIP) [18] aims to improve security by providing accountability on the network layer. AIP uses globally self-certifying unique end-point identifiers (EID) to identify and address the source and the destination of the connection, in addition to normal IP addresses. EIDs contain hashes of host's public keys that are communicating

within the network. AIP aims to prevent source address spoofing in the following way. If the router receives a packet from the unknown EID, the router will send a verification message back and the node will reply with a message signed by its private key. Since EID is hash of node's public key, this proves that the node owns a corresponding private key and thus has a right to use the EID. A similar method can also be used to authenticate domains.

Identity-based encryption and signature scheme (IBE) [24] allows a label, e.g., the user's e-mail address to be used as user's public key, simplifying the key distribution problem. However, IBE relies on a centralized entity called private key generator (PKG), which knows all private keys of its users. Therefore IBE can not be considered to be a suitable approach for securing large scale systems.

VII. Conclusions

This paper describes a complete security architecture for the clean slate data-oriented pub/sub networking. The solution secures both the rendezvous and forwarding planes, providing availability to all network functions.

Basically, we have provided an extensive and strong set of security mechanisms to protect the network against various attacks. More detailed study is required into policies that can most efficiently utilize these mechanisms. For example, when should malicious users be dropped from the network, or how to efficiently manage long-term identifiers.

So far we have experimented with a partial prototype and we plan to fully implement our system in the future. Since making drastic changes to the Internet architecture at once is not realistic, we also plan to investigate how to gradually deploy the system.

References

[1] K. Visala, D. Lagutin, and S. Tarkoma, "LANES: An Inter-Domain Data-Oriented Routing Architecture," in *ReArch'09*, Dec. 2009.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. Plass, N. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of ACM CoNEXT 2009*, 2009.

[3] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *Proceedings of ACM SIGCOMM 2007*, Kyoto, Japan, Aug. 2007.

[4] C. Wang, A. Carzaniga, D. Evans, and A. L. Wolf, "Security issues and requirements for Internet-scale publish-subscribe systems," in *Proceedings of the 35th annual Hawaii international conference on system sciences (HICSS'02)*, Hawaii, USA, 2002.

[5] P. Ganesan, K. Gummadi, and H. Garcia-Molina, "Canon in G Major: Designing DHTs with Hierarchical Structure," in *ICDCS'04*. IEEE Computer Society, 2004, pp. 263–272.

[6] V. S. Miller, "Use of elliptic curves in cryptography," in *CRYPTO '85: Proceedings of the Advances in Cryptology*, August 1985.

[7] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for key management - part 1: General, nist special publication 800-57," National Institute of Standards and Technology, Tech. Rep., Mar. 2009.

[8] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *ACM Communications*, vol. 13, no. 7, pp. 422–426, 1970.

[9] D. Lagutin, "Redesigning internet - the packet level authentication architecture," Licentiate's Thesis in Computer Science, Helsinki University of Technology, Espoo, Finland, 2008.

[10] J. Forsten, K. Järvinen, and J. Skyttä, "Packet level authentication: Hardware subtask final report," Helsinki University of Technology, Tech. Rep., 2008. [Online]. Available: http://www.tcs.hut.fi/Software/PLA/new/doc/PLA_HW_final_report.pdf

[11] R. Rivest, "S-Expressions (draft-rivest-sexp-00.txt)," Network Working Group, Tech. Rep., May 1997. [Online]. Available: http://people.csail.mit.edu/rivest/Sexp.txt

[12] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure routing for structured peer-to-peer overlay networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, pp. 299–314, 2002.

[13] J. R. Douceur, "The Sybil Attack," in *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. Lecture Notes in Computer Science, vol. 2429. London, UK: Springer-Verlag, 2002, pp. 251–260.

[14] N. Daswani, "Denial-of-service (dos) attacks and commerce infrastructure in peer-to-peer networks (draft)," Ph.D. dissertation, Stanford, Jan. 2005.

[15] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, "Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing," UC Berkeley, Tech. Rep., Apr. 2001.

[16] P. Jokela, A. Zahemszky, C. Esteve, S. Arianfar, and P. Nikander, "LIPSIN: Line speed publish/subscribe inter-networking," in *Proceedings of ACM SIGCOMM 2009*, Barcelona, Spain, Aug. 2009.

[17] C. Esteve, P. Jokela, P. Nikander, M. Srel, and J. Ylitalo, "Self-routing Denial-of-Service Resistant Capabilities using In-packet Bloom Filters," *Proceedings of European Conference on Computer Network Defence (EC2ND)*, 2009.

[18] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker, "Accountable internet protocol (AIP)," in *Proceedings of the ACM SIGCOMM 2008*. Seattle, USA: ACM, 2007, pp. 339–350.

[19] "Netfpga," 2009. [Online]. Available: http://www.netfpga.org/

[20] L. I. Pesonen and J. Bacon, "Secure event types in content-based, multi-domain publish/subscribe systems," in *Proceedings of the 5th international workshop on Software engineering and middleware*, 2005, pp. 98–105.

[21] R. Merkle, "Secrecy, authentication, and public key systems." Ph.D. dissertation, Department of Electrical Engineering, Stanford University, 1979.

[22] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The tesla broadcast authentication protocol," *Cryptobytes*, vol. 5, no. 2, pp. 2–13, 2002.

[23] T. Heer, S. Gtz, O. G. Morchon, and K. Wehrle, "Alpha: An adaptive and lightweight protocol for hopbyhop authentication," in *Proceedings of ACM CoNEXT 2008*, 2008.

[24] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of CRYPTO*, vol. 84. Springer, 1984, pp. 47–53.