PrIXP: Preserving the Privacy of Routing Policies at Internet eXchange Points

Marco Chiesa*, Roberto di Lallo†, Gabriele Lospoto†,
Habib Mostafaei†, Massimo Rimondini†, Giuseppe Di Battista†
*Université catholique de Louvain, IP Networking Lab
†Roma Tre University, Department of Engineering
marco.chiesa@uclouvain.be {dilallo,lospoto,mostafae,rimondin,gdb}@ing.uniroma3.it

***Abstract*—Internet eXchange Points (IXPs) serve as landmarks where many network service providers meet to obtain reciprocal connectivity. Some of them, especially the largest, offer route servers as a convenient technology to simplify the setup of a high number of bi-lateral peerings. Due to their potential to support a quick and easy interconnection among the networks of multiple providers, IXPs are becoming increasingly popular and widespread, and route servers are exploited increasingly often. However, in an ever-growing level of market competition, service providers are pushed to develop concerns about many aspects that are strategic for their business, ranging from commercial agreements with other members of an IXP to the policies that are adopted in exchanging routing information with them.**

Although these aspects are notoriously sensitive for network service providers, current IXP architectures offer no guarantees to enforce the privacy of such business-critical information. We re-design a traditional route server and propose an approach to enforce the privacy of peering relationships and routing policies that it manages. Our proposed architecture ensures that nobody, not even a third party, can access such information unless it is the legitimate owner (i.e., the IXP member that set up the policy), yet allowing the route server to apply the requested policies and each IXP member to verify that such policies have been correctly deployed. We implemented the route server and tested our solutions in a simulated environment, tracking and analyzing the number of exchanged control plane messages.

I. Introduction

Organizations that offer Internet-based services (Internet Service Providers, Content Delivery Networks, etc.) join the Internet eXchange Points (IXPs) in order to quickly and easily reach a number of other parties networks, and gain the level of connectivity they need [1]. However, such organizations are usually concerned with business-critical aspects for which IXPs do not currently provide any technical solutions. These aspects include, among the others: (i) privacy of the peering relationships, which are an evidence of the existence of commercial agreements; (ii) privacy of routing policies, which determine what kind of traffic can flow between peering partners; (iii) security of the network infrastructure (links, devices), that might be traversed by sensitive traffic.

Work partially supported by the MIUR PRIN national research project 2012C4E3KT ("AMANDA – Algorithmics for MAssive and Networked DAta) and by the EU FP7 project "Preemptive – Preventive Methodologies and Tools to Protect Utilities", grant no. 607093. This research is (in part) supported by European Unions Horizon 2020 research and innovation programme under the ENDEAVOUR project (grant agreement 644960).

Currently, IXPs offer a very useful service, called Route Server (RS). An RS allows each member connected to an IXP to easily exchange traffic with other members by establishing a peering session with the RS, instead of having one peering with each other member he wants to be connected to. Peering sessions are handled by the Border Gateway Protocol (BGP), the standard interdomain routing protocol. Surely, this functionality significantly reduces the effort needed by the IXP members to connect to the Internet.

Ensuring the privacy and correctness of Internet peering policies is a desired requirement for many Internet entities as this information reflects business relationships, such as commercial agreements, which must comply with stringent Service Level Agreements (SLAs). Very often, RS functionalities are mainly leveraged by small providers and Content Delivery Networks (e.g. [2], [3], [4], [5]) since these players have strong interests in connecting to many IXP members by just setting up a single BGP peering with RS. On the other hand, big Internet players, with very few exceptions (e.g. Google [2]), tend to not have BGP peerings with an RS. We argue that this trend is the result of exposing an IXP member to a potential violation of privacy in terms of BGP policies when peering with an RS. In fact, each peering policy would be stored within appropriate data structures at the RS and, potentially, these can be altered by a malicious software. As a result, most Tier-1 ISPs require their peers to sign Non Disclosure Agreements (NDAs) when peering with them [6].

In this paper, we present PrIXP, an RS system that improves both the privacy guarantees of confidential peering information and the security of the RS. Our key idea is to prevent the RS from locally storing any BGP policies. Instead, the RS queries routing policies in on-demand manner by means of a second communication channel that we instantiate between the RS and each IXP member. Namely, each time the RS performs a routing operation, it leverages this extra channel to retrieve from each member its routing policies such as the set of member neighbors that should receive certain routing information and the local preference over routes of each member. To guarantee the correct execution of the BGP routing protocol at the RS, we leverage Intel proprietary Software Guard eXtensions (SGX) technology [7], which allows external entities to attest that a remote software has not been tampered by a malicious attacker. Finally, to enable incremental deployment, we discuss a BGP compatible mechanism that can be used in place of the extra channel, thus requiring no hardware modifications at the IXP member side.

The rest of the paper is organized as follows. In Sec. II, we provide an overview of a common real-world architecture of a route server deployed inside IXPs. In Sec. III, we describe our system in detail, presenting a complete example of an interaction between the route server and the IXP members connected to it. In Sec. IV, we address the security issues associated with peering with a traditional RS by describing our solution for allowing any IXP member to check the integrity of the RS. In Sec. V, we evaluate our system by using a real-world trace of BGP updates from one of the largest IXP worldwide. In Sec. VI, we review the most relevant contributions related to Internet routing privacy and security. Finally, we draw conclusions and future work in Sec. VII.

II. Background: Route Server Architecture

In this section, we describe the typical architecture of a RS service offered to the members of an IXP (e.g. [8]). To the best of our knowledge, many large IXPs such as DE-CIX, AMS-IX, and NYIIX are currently using RS implementations based on that architecture.

Before entering into the details, we introduce terminology and definitions that will be largely used throughout the paper. We define the *RS-software* as the piece of software implementing the RS functionality and the *RS-machine* as the physical hardware that runs the RS-software. We also define the *peering LAN* as the local network managed by the IXP where its members connect and establish BGP peerings among themselves and with the RS-software.

In a standard scenario, each member of an IXP establishes a number of *bi-lateral* BGP peerings with all the members with whom it has agreed to exchange network traffic for certain IP prefix destinations. Such bi-lateral peerings usually correspond to commercial agreements between the involved parties. In contrast to this approach, many IXPs provide RS services as a convenient alternative for their members to simplify the setup of peerings while optimizing the operation of the BGP control plane. Indeed, RSes reduce the configuration effort required by network operators to join and manage many *bi-lateral* BGP sessions at large IXPs, since having a single BGP peering with the RS is enough to be connected with the other members.

We now describe the set of operations that an IXP member should perform in order to make use of RS services. First, the IXP member establishes a single BGP peering towards the RS-machine with the RS-software, which is responsible for forwarding any BGP announcements according to the routing policies configured by the members. The above scenario is denoted as a *multi-lateral* peering, where the RS acts as the center of a star topology where the members are called *clients*.

The architecture of an RS-software is shown in Fig. 1. In this figure, AS1, AS2, AS3, and AS4 are members of the IXP, each of them connected to the IXP using a BGP-speaking router, where the dashed lines labeled B1, B2, B3, and B4 represent BGP peering sessions. Each of these routers

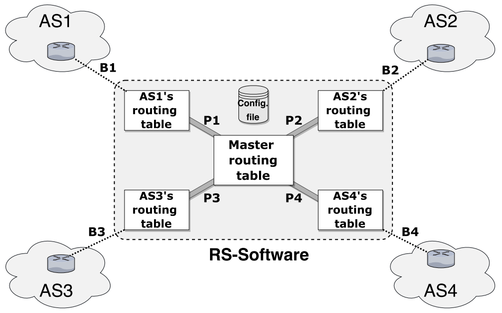

Fig. 1. Reference architecture of route server.

independently keeps a routing table that stores the IP prefixes coming from its own network, as well as those received from its multi-lateral peers through the RS. The rounded dashed box labeled "RS-software" represents an instance of the RS routing software, where the contents of the box depict the most important data structures that are maintained by the software and the channels used to move data among these structures. We now describe each basic component represent in the figure.

Tables. The basic data structures maintained by an RS are *BGP tables*. A BGP table contains a set of routing entries, each of them consisting of an IP prefix and the BGP message announcing that prefix. Multiple entries for the same prefix may exist, though only one of them is marked as the best one that should be propagated to the other members. For each member, an RS-software keeps a distinct table that stores all the routes that are announced towards that client from other clients. In order to support the exchange of routing information among these tables, the RS also maintains a single *master* table, which usually aggregates all the routes received from all the client-specific tables.

Protocols. The RS software leverages different communication channels for transferring information among tables, called *protocols*. BGP routes are exchanged between a client and one of the member-specific tables inside the RS-software through a BGP session (lines B1, B2, B3, B4 in Fig. 1). The routes learned from these sessions can then be propagated between the different tables using a RS-specific protocol, which corresponds to the links among the BGP tables (thick lines P1, P2, P3, and P4 in Fig. 1).

Filters. In order to support arbitrary routing policies, it is also possible to define filters. A *filter* is typically a fragment of code, possibly written in a specific programming language, that supports evaluation of arithmetic expressions, conditional statements, etc. Filters are applied on each BGP announcement ever time they are exchanged through BGP sessions or RS-specific protocols. A filtering operation can have three possible outcomes: (1) forwarding the announcement, (2) modifying some attributes in the announcement before forwarding it, and (3) dropping the announcement. Filters can be statically configured within the RS software by the IXP operators. This practice is commonly adopted for limiting the risk of IP-prefix hijacking. The common way to perform filtering is encoding the set of members to whom a routing announcement must be sent via specific BGP attributes that are attached to the announcement itself, i.e., via BGP communities, where each BGP community simply consists of a pair (x, y) of values. We define a *whitelist export policy* as the set of members ($\mathsf{AS\_1}, \ldots, \mathsf{AS\_N}$) encoding all members that are allowed to receive a BGP announcement. A whitelist is expressed by a sequence of community values starting with a special community ($\mathsf{0} : \mathsf{IXP\_id}$), followed by a sequence of other community values ($\mathsf{IXP\_id}, \mathsf{AS\_i}$), for each $i = 1, \ldots, N$ representing all members to which forward the announcement. In the same way, we define a *blacklist export policy* as a sequence of community values encoding the set of ASes ($\mathsf{AS\_1}, \ldots, \mathsf{AS\_N}$) that should not receive a BGP announcement. A blacklist always starts with a special community ($\mathsf{IXP\_id} : \mathsf{IXP\_id}$) and it is followed by a pair $(0, \mathsf{AS\_i})$, for each member that is denied to receive the announcement.

Best Route Selection and Propagation. Unless filters enforce restrictions, the adoption of a specific internal protocol, as explained before, causes all BGP routes to be copied between the tables it links, retaining all their attributes and including non-best routes. Each best route for a member is computed using the information gathered in its specific member routing table. This strategy allows IXP operators to overcome the well-know problem known as *path hiding*, which arises whenever filters are applied [9], [8]. This is a well-known problem that might affect members if the RS-software acts as a standard BGP router, where a single master route table is used to collect all the route announcements and to compute a unique best route for all the customers. For example, consider the case in which there are four members ($\mathsf{AS1}$, $\mathsf{AS2}$, $\mathsf{AS3}$, and $\mathsf{AS4}$) connected to a RS-machine through a multi-lateral peering. An IP prefix π is announced by $\mathsf{AS1}$ and $\mathsf{AS2}$ and the latter one defines a restricted policy that prevents $\mathsf{AS3}$ to receive the announcement containing π. Also, suppose that the RS-software runs the best route process only considering the routes contained in the master table and that computation selects the route passing through $\mathsf{AS2}$ as the best one. In this case, this route is only advertised to $\mathsf{AS4}$, leaving $\mathsf{AS3}$ without any route towards π, even though a route passing through $\mathsf{AS1}$ exists. Breaking down the master table into per-member tables makes possible to run independent best route computation on each member table, preventing the above situation to happen. Although the BGP configuration language allows routes to be ranked based on the *local preferences* of each member, today's RSes do not support this mechanism and the best route is computed based on a global ranking, as defined in [10].

III. Enforcing Privacy of Routing Policies

In this section, we describe how PrIXP improves the level of privacy for the members' routing policies within an RS-machine. In our system, each member can easily leverage the RS's functionalities (e.g. BGP routes dispatch based on export policies and local-preference tools) while minimizing the risk of leaking any confidential information. Our system does not propose an entirely new cryptographic protocol, but leverages well-established techniques (e.g., TLS, SGX) to secure channels and performing remote attestation. Those techniques can be replaced by any equivalent technology.

We observe that current RSes designs (described in Sec. II) require IXP members to disclose their export policies. In fact, any peering relationships among the IXP members can be reconstructed by simply looking at the client-specific routing tables stored in memory or on disk. In fact, the table of a member $\mathsf{AS}x$ contains all the routes received by other ASes, thus revealing what are the export policies of each member towards ASx. Moreover, any enhanced RS service that allows the IXP member to rank their available routes based on their specific local preferences would raise additional privacy concerns. In fact, such services would require each member to disclose their ranking policies to the IXP.

We now describe the security assumptions and the threat model on which our system is based. First, we assume that the attacker does not have visibility of data traffic. Namely, an attacker cannot eavesdrop the packets sent through the peering LAN of the IXP in order to infer the peering relationship among the members. Second, we assume that the attacker operates on the RS-machine during a short time interval in which he tries to take a snapshot of as much information as possible from the content stored in the route server system, possibly tampering the RS-software itself.

Our system is based on the following principles. The only information that is stored within the PrIXP RS is the one needed to maintain the established BGP sessions with the IXP members and, for each announced prefix, the set of members that have a route towards it. The routing policies of the IXP members are never permanently stored by the RS-software inside the RS-machine so as to minimize the risk of privacy breaching. In contrast, these policies are asked to the members in response to the reception of a BGP announcement that has to be dispatched. We make use of an extra communication channel for retrieving this information, which can be set up using standard techniques (e.g. SSL/TLS). We observe that, in order to minimize the modification required at the member side, it would be worth to investigate how to implement this channel by tweaking the BGP protocols. The idea is to leverage *Conditional Route Advertisement*, a BGP route dissemination feature that allows to conditionally announce one or more prefixes upon the reception of some specific routes. Such a feature is currently supported by important vendors, as shown in ([11], [12]).

The extra communication channel is used by the PrIXP RS to query each member for the following information: (i) the export policies of a routing announcement (e.g. the set of members to whom a route should be propagated) and (ii) the local preferences over routes of each BGP member that is entitled to receive a BGP message. We now provide a detailed description of the operations performed by the PrIXP.

A Complete Example. A simplified scenario that we use to illustrate how our system works is depicted in Fig. 2. The RS-machine is placed in the middle of the drawing, while

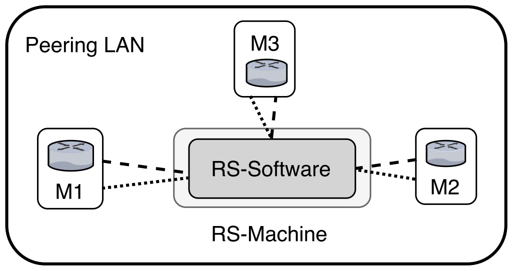

Fig. 2. The architecture of an IXP infrastructure.

the three members (M1, M2, and M3) are connected to the IXP physical infrastructure. For our convenience, we assume that M1, M2, and M3 are also the identifier of the three members, respectively. The rounded rectangle containing the whole drawing represents the peering LAN and we assume that the peering LAN consists of a single switch. Each dashed line represents a BGP session, whereas dotted lines represent the extra communication channel used by PrIXP to query each member. To make use of the RS's functionalities, each member establishes a BGP session with the RS-software. Each member can still establish bi-lateral BGP sessions with the other members as in traditional RSes.

Once all the BGP connections are established, members can use them to exchange routing information among each other. For instance, suppose that M1 and M2 send an announcement towards an IP prefix π to the RS-software, which is responsible for dispatching it according to the members' routing policies. Upon receiving this message, PrIXP asks M1 for the export-policy. Member M1 replies to this request by communicating a set of BGP communities encoding a policy that allows the RS-software to advertise the announcement to M3. After delivering the message, the RS-software stores in its memory that it received a route for π from M1, but it deletes any other additional information (e.g. the export policies and the BGP attributes contained in the announcement).

Now, suppose that also M2 sends an announcement towards π to the RS-software. When the RS-software receives that message, it checks whether there exist other routes announced towards π. Then, it asks each member that announced a route towards π (M1 and M2) for the export policies of their announcement using the extra communication channel. Member M1 communicates again that its announcement must be announced to M3, while M2 instructs the RS-software to propagate its message to both M1 and M3. Upon receiving the export policies, the RS-software knows which routes can be exported to each member. In order to select the best one, the RS-software asks each member with at least two available routes for the local-preference of each route announcement. In our case, the RS-software asks M3 to provide the ranking over the routes announced by M1 and M2. Once M3 provides its local preference, the best route is sent to M3. As for M2, the only route that is available to be exported to it is the one through M3, which is then propagated accordingly. Note that, this last step is performed over the BGP peering. After that computation, the RS-software discards all the information used to propagate the routes, except for the mapping between routes and members who announced them. This operations allows us to minimize the risk of leaking routing policies whenever an attacker can observe the state of the RS-software for a short interval of time. Note that having a single BGP decision process for each member makes our RS-software not affected by the *path hiding* problem.

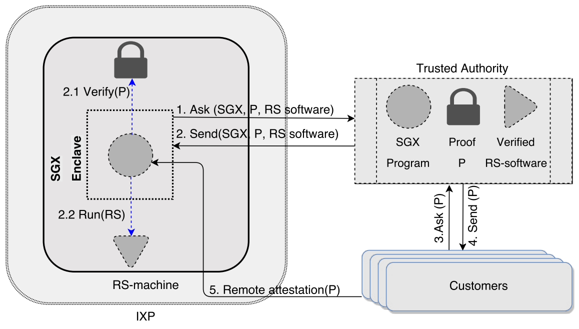

Fig. 3. Architecture for checking the integrity of the RS software.

IV. Discussion on Security Issues

In this section, we describe some security considerations, addressing the problem of how a member can verify that the RS-software has not been tampered or replaced by another malicious software. To minimize the risk of leaking confidential information, we assumed in Sec. III that each member is connected to a trusted execution of the PrIXP RS-software. Under our threat model, we assume that the attacker may quickly replace the RS-software to collect confidential information that can be read by the attacker next in the future. For this reason, we also define an RS security architecture, depicted in Fig. 3, which is based on recent advancements in remote attestation protocols. A trusted authority issues a certified version of the RS-software that each member can verify on its local machine, implemented according to the description of the PrIXP system in the previous system, represented by a triangle in the picture. Unfortunately, to allow all the IXP members to be able to check that at any time the RS-software is behaving as expected, having just a certified version of the RS-software is not enough, because a member does not have any tools to attest at any time that exactly the certified version of the software is running. Indeed, an attacker can suitably replace that certified version of the RS-software. To solve this problem, we rely on the recent advancements on *Remote Attestation*, which allows changes to the RS-software to be detected by authorized parties. Intel Security Guard eXtension (SGX) [7] is an example of a technology that allows programmers to implement remote attestation procedures.

Each SGX program needs a proof to be executed on a SGX-enabled machine. In our architecture, the trusted authority provides to the RS-machine an SGX program and a proof P, respectively depicted by the circle and the lock in Fig. 3.

The integrity check works as follows. First, the RS-machine, which has an SGX processor, sends to trusted authority a request to obtain the RS-software, the SGX program and the proof P. This is represented as the step 1 in the figure. Upon receiving this request, the trusted authority sends back to the RS-machine the RS-software, the SGX program and the proof P. This is step 2 in the figure. At this point, the RS-machine owns all the necessary pieces to correctly run the verified RS-software. This task is accomplished by the SGX program that can run if and only if the proof P has been verified (step 2.1). In order to guarantee that the RS-machine will run the correct version of the RS-software, the SGX-program will check that the hash of the RS-software to be executed corresponds to the one that is hard coded in the SGX-program retrieved from the central authority (step 2.2). At this point, whenever a member wants to check the integrity of the RS-software, it asks the trusted authority for the proof P, which is denoted as step 3 in the figure. Upon receiving the proof P, a member performs a remote attestation against the SGX program running at the RS-machine by using the proof P (steps 4 and 5 in the picture).

Since the proof P used by a member for the remote attestation is the same used by the SGX program at the RS-machine to run the RS-software, the operation succeeds. If an attacker aims at replacing the RS-software, he must also replace the SGX program, otherwise the hash-check would not allow him to run its own RS-software. This implies that a new proof P must be provided to the SGX-machine in order to run the malicious SGX-program. At this point, if the attackers succeed in running its malicious SGX-program and RS-software, the remote attestation performed by any member using the proof P would fail, as the SGX-machine would alert the user the SGX program is not the legitimate one.

In this paper, we do not propose any new cryptographic protocol, but we leverage well-established techniques (e.g. TLS for the extra communication channel and SGX for remote attestation). Those techniques can be replaced by any equivalent technologies without altering the PrIXP functionalities.

V. Experiments

To assess the effectiveness of PrIXP, we simulated our system (available at [13]) using a trace of BGP updates from one of the largest IXP worldwide with several hundreds of members whose name cannot be disclosed in this paper. Our simulation aims at estimating how much overhead our methodology introduces in terms of BGP control plane messages. We do not measure CPU overhead or memory utilization, since we do not expect both of them to be a bottleneck as PrIXP only uses simple access to data structures and stores less information than traditional RSes.

First, we implemented a prototype RS-software written in Python, as depicted in Fig. 4, including a decision process acting according to the route dispatching mechanism described in Sec. III. To easily manipulate BGP messages within the RS-software, we relied on ExaBGP [14], a software tool for easily interfacing and managing BGP sessions in a convenient JSON format. The input of our simulation is a dump of all the routes announced inside a big European IXP in a one hour time interval. We ran two different experiments: the first one using a traditional RS-software that does not guarantee any privacy, and the second one using PrIXP. During each experiment, we collected the number of exchanged messages to quantify the communication overhead due to the extra channel communication. To put ourselves in the worst-case condition, we assumed that each member is willing to send its route announcements to any other member.

The percentage of members that received at least a certain amount of BGP announcements from the RS-software is depicted in Fig. 5. The red line refers to the standard RS-software, whereas the black one represents the CDF for our methodology. We see that in PrIXP around 95% of the members received roughly 5000 BGP announcements more that with respect to the standard RS-software, which is an overhead by a factor of 1.5. We argue that this amount of overhead is affordable for a member, considering the time interval taken into account. The number of messages sent by each member to the RS-software is depicted in Fig. 6. Note that the red line is now very close to the leftmost part of the graph, showing that only a few members announce many

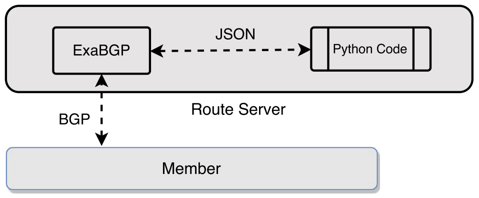

Fig. 4. Architecture of our RS-software prototype implementation.

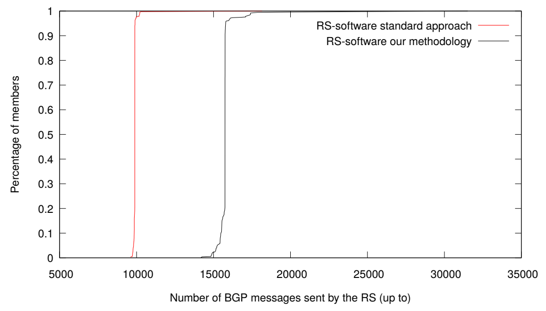

Fig. 5. CDF of the number of messages issued by the RS-software.

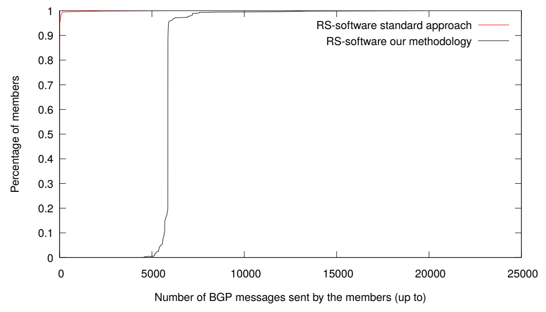

Fig. 6. CDF of the number of messages issued by members.

routes, while the vast majority of the IXP members are not involved in sending many BGP routes. In this case, we observe a significant increase in terms of number of messages, since that amount includes the messages exchanged on the extra communication channel in order to ensure privacy at the RS-software, which is not guaranteed in the standard approach. We argue that it is due to the fact that the PrIXP does not store in memory any routing information at the RS, thus forcing the members to send them when required.

To allow members to verify the integrity of the RS-software, we used *OpenSGX* ([15], [16]), an open source implementation of SGX. This experiment aims at verifying that the remote attestation mechanism described in Sec. IV behaves as expected. We produced a hash value of the PrIXP implementation. Then, we wrote an SGX program that executes the RS-software only if the hash of the RS-software corresponds to the one of the precomputed hash. To generate the proof P of the SGX program, we used a key issued by the trusted authority, according to Fig. 3. We ran the SGX program on a virtual machine acting as RS-machine. After checking the checksum of the RS-software, our SGX program successfully executed it. At that point, we tried to perform a remote attestation from an external client towards the SGX program running on the RS-machine. To do that, we provided the trusted proof P from the external machine to the SGX one, and in that case, the remote attestation succeeded. After that, we altered the code of the RS-software, and the SGX program detected the change as it did not run the malicious RS-software. As the final step, we executed on the RS-machine a malicious SGX program with a proof P' generated using a malicious key, with an altered version of the RS-software. The member detects that the RS-software was tampered since the remote attestation fails.

VI. Related Work

In this section, we overview the most relevant work to ours along two dimensions: (i) securing the Internet routing computation and (ii) preserving the privacy of the routing policies on the Internet.

Security of Internet routing. Several attempts have been made by the Internet community in order to secure the Internet routing from malicious activity such as IP-prefix hijacks and similar attacks. The set of techniques developed to curb these malicious activities range from Resource Public Key Infrastructure (RPKI) [17], which is used to verify whether the originator of a BGP announcement is the legitimate one, to Secure BGP (S-BGP) [18], which allows any entity to verify the authenticity and authorization of BGP control traffic. We note that, beyond large-scale deployment issues with these techniques, none of them can actually be used to guarantee the IXP network will correctly propagate the BGP announcements. The IXP operator can still (i) do not propagate a BGP route or (ii) select any of its known routes as the best one. Nevertheless, an implementation of a RPKI-based route server is in [19].

Several efforts have been made to improve the level of security offered by RPKI and S-BGP. These efforts include the most closely related work to ours, SPIDER [20], which devised a distributed mechanism that allows the peers of a network to verify a number of nontrivial properties of its interdomain routing decisions (such as adherence to the BGP protocol) without revealing any additional information (beyond those revealed by the underline protocol, i.e., BGP). When casting this mechanism in the IXP setting, SPIDER allows each IXP member to verify that the IXP is not deviating from the BGP protocol (i.e., sending non-best routes), but it requires the IXP members to disclose their routing policies to the IXP operator.

Privacy of Internet routing. In [21] and [22], Secure MultiParty Computation (SMPC) techniques have been used in order to compute Internet routing paths without revealing to any party the routing policies of the Internet entities. SMPC is a branch of cryptography that studies the problem of computing a function over their inputs while keeping those inputs private. As the authors themselves recognize [21], the main drawback of using SMPC lies in the inherent difficulty of scaling it to a large number of participants, as the computational and communication complexity easily becomes a bottleneck, especially when the SMPC function is required to be robust against malicious attackers.

Kim et al. [23] make extensive use of Intel SGX to preserve the privacy of ISPs' policies and to guarantee the correct propagation of BGP announcements. SGX is a proprietary hardware-based mechanism that allows programmers to create *enclaves* of memory by means of special processor's instructions. In order to limit our dependency with a proprietary building block, we use SGX to remote attestation only, providing the privacy of routing policy in a distributed manner.

VII. Conclusions and Future Works

During the last decade, IXPs emerged as economically advantageous solutions for interconnecting multiple Internet entities. While RS services have been deployed at IXPs to ease the operators from the burden of managing hundreds of BGP sessions, the usage of such services have been hindered by the privacy concerns regarding the disclosure of the members' routing policies to external commercial parties such as the IXP.

In this paper, we designed PrIXP, an RS service that allows to redistribute BGP routing information according to the import/export policies specified by the IXP members while minimizing the risk of leaking that information to any curious or malicious entity. We demonstrated that PrIXP has little message overhead compared to traditional non-secure RSes and it requires only minor modifications at the members' side.

In the next future, we plan to pursue the following directions. First, we intend to improve our prototype implementation, aiming at reducing the control plane overhead introduced by the current version and assessing the computational overhead in our system. Second, we will extend our experimental setup to in order to gather information about other relevant metrics such as the time spent by a member to receive the legitimate routes. Finally, we will devote our efforts towards eliminating any hardware modification at the members side in order to ease the deployment of PrIXP at any IXP by tweaking the BGP protocol.

References

[1] M. Di Bartolomeo, G. Di Battista, R. di Lallo, and C. Squarcella, "Is it really worth to peer at ixps? a comparative study," in *Proc. 20th IEEE Symposium on Computers and Communication (ISCC 2015)*, 2015, pp. 421–426.

[2] AMS-IX, "Amsterdam internet exchange point members," Sept 2016. [Online]. Available: https://ams-ix.net/connected_parties

[3] LINX, "London internet exchange point route server members," Sept 2016. [Online]. Available: https://www.linx.net/tech-info-help/route-servers

[4] FRANCE-IX, "France internet exchange point members," Sept 2016. [Online]. Available: https://www.franceix.net/en/france-ix-paris/members-in-paris/

[5] MIX, "Milan internet exchange point members," Sept 2016. [Online]. Available: http://www.mix-it.net/index.php?option=com_content&view=article&id=85&Itemid=84&lang=it

[6] D. Peering, "Peering policy clauses collected from 28 companies," Internet Peering Workshop, 2012. [Online]. Available: http://drpeering.net/workshops/presos/16%20Peering-Policy-Clauses.pdf

[7] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar, "Innovative instructions and software model for isolated execution," in *Proceedings of the 2Nd International Workshop on Hardware and Architectural Support for Security and Privacy*, ser. HASP '13. New York, NY, USA: ACM, 2013, pp. 10:1–10:1.

[8] P. Richter, G. Smaragdakis, A. Feldmann, N. Chatzis, J. Boettger, and W. Willinger, "Peering at peerings: On the role of ixp route servers," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14. New York, NY, USA: ACM, 2014, pp. 31–44. [Online]. Available: http://doi.acm.org/10.1145/2663716.2663757

[9] E. Jasinska, N. Hilliard, R. Raszuk, and N. Bakker, "Internet exchange BGP route server," IETF draft-ietf-idr-ix-bgp-route-server-10, Apr 2016.

[10] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271 (Draft Standard), Jan. 2006.

[11] C. Support, "Configuring and verifying the bgp conditional advertisement feature," Sep 2016. [Online]. Available: http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/16137-cond-adv.html

[12] J. TechLibrary, "Configuring conditional installation of prefixes in a routing table," Apr 2016. [Online]. Available: http://www.juniper.net/techpubs/en_US/junos13.3/topics/example/conditional-prefix-installing-configuring.html

[13] R. T. University, "Computer networks research groups," Sept 2016. [Online]. Available: https://bitbucket.org/rdl87/prixp/src

[14] Exa-Networks, "Exabgp," Sep 2016. [Online]. Available: https://github.com/Exa-Networks/exabgp

[15] P. Jain, S. Desai, S. Kim, M.-W. Shih, J. Lee, C. Choi, Y. Shin, T. Kim, B. B. Kang, and D. Han, "Opensgx: An open platform for sgx research," in *Proceedings of the Network and Distributed System Security Symposium, San Diego, CA*, 2016.

[16] OpenSGX, "Opensgx: An open platform for intel sgx," Sep 2016. [Online]. Available: https://github.com/sslab-gatech/opensgx/

[17] R. Bush and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol," IETF RFC 1997, Jun. 2013.

[18] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol (s-bgp)," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 582–592, April 2000.

[19] K. Kim and Y. Kim, "The security appliance to bird software router," in *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication*. ACM, 2014, p. 37.

[20] M. Zhao, W. Zhou, A. J. Gurney, A. Haeberlen, M. Sherr, and B. T. Loo, "Private and verifiable interdomain routing decisions," in *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, ser. SIGCOMM '12. New York, NY, USA: ACM, 2012, pp. 383–394. [Online]. Available: http://doi.acm.org/10.1145/2342356.2342434

[21] D. Gupta, A. Segal, A. Panda, G. Segev, M. Schapira, J. Feigenbaum, J. Rexford, and S. Shenker, "A new approach to interdomain routing based on secure multi-party computation," in *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*, ser. HotNets-XI. New York, NY, USA: ACM, 2012, pp. 37–42. [Online]. Available: http://doi.acm.org/10.1145/2390231.2390238

[22] M. Chiesa, D. Demmler, M. Canini, M. Schapira, and T. Schneider, "Towards securing internet exchange points against curious onlookers," in *Applied Networking Research Workshop (ANRW 2016)*, 2016.

[23] S. Kim, Y. Shin, J. Ha, T. Kim, and D. Han, "A first step towards leveraging commodity trusted execution environments for network applications," in *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, ser. HotNets-XIV. New York, NY, USA: ACM, 2015, pp. 7:1–7:7.