SIXPACK: Securing Internet eXchange Points Against Curious onlooKers

Keywords : Routing; Internet exchange points; Privacy.

Marco Chiesa, Daniel Demmler, Marco Canini, Michael Schapira, Thomas Schneider

Abstract* – *The growing relevance of Internet eXchange Points (IXPs), where an increasing number of networks exchange routing information, poses fundamental questions regarding the privacy guarantees of confidential business information. To facilitate the exchange of routes among their members, IXPs provide Route Server (RS) services to dispatch the routes according to each member's peering policies. Nowadays, to make use of RSes, these policies must be disclosed to the IXP. This state of affairs raises privacy concerns among network administrators. We design SIXPACK, a RS service that leverages Secure Multi-Party Computation techniques to keep peering policies confidential, while maintaining the same functionalities as today's RSes. We assess the effectiveness and scalability of our system using traces of data from one of the largest IXPs in the world.

Protecting the privacy of sensitive business data on the Internet is a topic that is subject to ever-growing attention in a highly-connected, insecure world. We focus on the goal of preventing the leakage of business policies in Internet routing. With the advent of Internet eXchange Points (IXPs) as the new physical convergence points for Internet traffic, new privacy concerns arise. This is because IXPs offer centralized Route Server (RS) services for ranking, selecting, and dispatching routes to their member networks. However, to benefit from the centralized services, IXP members need to divulge private information, such as business peering relationships to the IXP.

The SIXPACK route dispatcher system. We propose a practical solution for protecting confidential peering policies of the IXP members, i.e., the specification of what BGP routes a member is willing to announce to other members, by executing today's route server services on critical data via secure multi-party computation (SMPC) [1]. Under SMPC, the computing entities do not gain visibility into neither the input nor the outputs of the computation. We leverage state-of-the-art accomplishments in SMPC to design SIXPACK, the first IXP route server service for ranking, selecting, and dispatching BGP routes without leaking any confidential business peering information. In our design, two non-colluding computing parties, called Route Server 1 and Route Server 2, carry out the dispatching of BGP routes (see Figure 1). We consider two different route dispatch approaches, which differ in the number of routes that are exported to the IXP members:

- *SINGLE*: Exporting only the best route. In the SINGLE dispatch approach, SIXPACK collects BGP announcements from all the IXP members, computes the best exportable route for each member, and dispatches to each member its selected route.
- *ALL*: Exporting all permissible routes. In the ALL dispatch approach, SIXPACK relays all exportable BGP announcements between its members. That is, SIXPACK performs route filtering so as to enforce members' export policies, but does not select best routes for its members.

To clarify our security assumptions, our threat model focuses on parties in the SMPC computation that have a perfect view of all BGP routes announced through the IXP but do not monitor the actual flow of traffic. We assume that both parties adhere to the protocol but attempt to infer as much information as possible about the private inputs (i.e., export policies) of the IXP members. Our goal is to prevent each of the parties from learning anything about these private inputs.

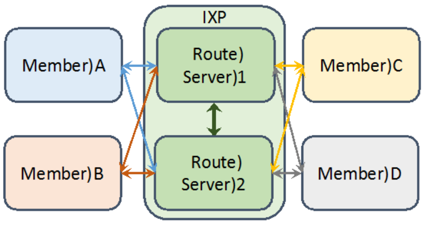

Figure 1: Conceptual overview of SIXPACK.

Implementation and evaluation. We implemented a fully functional prototype for SIXPACK. For implementing the SMPC part of the system, i.e., the two RSes components shown in Fig.1, we use the ABY framework [2]. ABY is implemented in C++ and provides low-level primitives for building SMPC functions. The computation of an SMPC function consists of an offline phase, which can be precomputed, and an online one, which depends on the real inputs. For implementing the rest of the system, i.e., the distribution and processing of all the BGP update information among RSes and IXP members, we used Python. We assess our system using a trace of BGP updates from one of the largest IXPs worldwide, with more than 600 members. Our results highlight the following:

- While SMPC is (as expected) the costliest part performance-wise, our results establish that the online phase is even at worst below 20 ms. The worst setup and online runtime we measured in our evaluation were 72ms and 19 ms, respectively for 32 inputs in the SINGLE case.
- Our unoptimized SIXPACK prototype is capable of processing BGP announcements in real time with a latency of 274 ms at the 99th percentile and negligible communication overhead, even without precomputing the offline phase of the SMPC.
- We measured the amount of communication that is required by SIXPACK during our evaluation of our datasets. We found that the average bandwidth requirements are: no more than 2.79 Mbps between the two route servers, no more than 70 Kbps between a member and the two route servers.

Future directions. We believe that future research should concentrate on extending the functionality of privacy-preserving RS services. One interesting direction is extending SIXPACK to receive as input (beyond members' export policies) also member's (private) local preferences over BGP routes and then running SMPC to select the best (exportable) path per member.

References

[1] A. C. Yao, "How to Generate and Exchange Secrets". In Foundations of Computer Science, pages 162-167, 1986.

[2] D. Demmler, T. Schneider, M. Zohner, "ABY: A Framework for Efficient Mixed-Protocol Secure Two-Party Computation", NDSS, 2015.