

IXP++: Towards Securing IXPs

*Muhammad Muqsit Nawaz, Faizan Safdar Ali, Dr. Muhammad Fareed Zaffar, Waqar Aqeel,
Usman Nazir
LUMS School of Science and Engineering Pakistan*

During the last few years, thousands of medium-to-large Internet Exchange Points (IXP) have emerged around the world. They operate a route server and offer its use as a free value-added service to their members. Original architects of IXPs and SDX (next generation Software Defined Exchange Points) paid less attention to accountability and security. Hence, this poses fundamental questions regarding the privacy concerns of confidential business information that is exchanged between members and Route Server (RS) services. Due to such reasons, Autonomous Systems (ASes) deter from providing intra-domain routing information, consequently resulting in an un-optimized traffic engineering. We have designed IXP++, a domain privacy-preserving security mechanism that employs homomorphic encryption and private set intersection in complex networks with SDXs to prevent leaking of crucial business information.

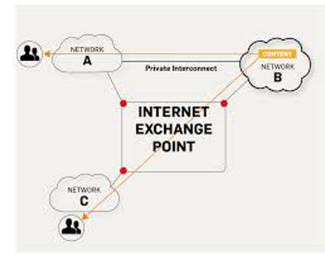
With newer Internet Exchange Points (IXPs) being established every day for mutual benefits of Autonomous Systems (ASes) involved, new privacy concerns arise. There are currently some 350+ Internet Exchange Points (IXPs) worldwide, and some of the largest and most successful IXPs have more than 500-600 members and carry as much traffic as some of the global Tier-1 ISPs. Due to being densely connected physical components in today's Internet, many of them have started to use route servers (RSes) as a free valued-added service to their members. An RS greatly simplifies routing for its members, that is, most members use an IXP's RS to establish multi-lateral peerings as compared to bi-lateral peerings.

Our main aim is to answer the question of verifiability i.e. the questions for example, whether the chosen route was the best one available, or whether it was consistent with the networks peering agreements, and privacy of crucial information (for example the route policies of the ASes) in SDXes. Since more and more IXPs are moving towards SDXes, we believe that implementing IXP++ in a software defined environment is the most viable option. When we want to exchange intra-domain routing information for the purpose of verifiability and an optimized route selection, privacy concerns arise and some network administrators avoid it for precisely this reason. For example a couple of Internet providers in the Nairobi central business district offered to host the IXP. The challenges were (1) how to choose between the two ISPs and (2) the high levels of dissatisfaction expressed by the other ISPs about having to trust a competitor to handle the IXP without seeking for itself an undue advantage.

In this paper, we are trying to propose solution to how the Centralized Controller in SDXs (acting as a Route Server) can allow the ASs connected to it to verify a number of nontrivial properties about inter-domain routing decisions without leaking confidential information. If all the properties hold, the peers learn nothing beyond what the inter-domain routing protocol already reveals; if a property does not hold, at least one peer can detect this and prove the violation. Furthermore, since we are ensuring privacy the two ASs who have peering agreements can provide information about their intra-domain routing which will result in better traffic engineering.

IXP++: TOWARDS SECURING IXPS

Muhammad Muzaffar Nawaz, Faizan Safdar Ali, Dr. Muhammad Fareed Zaffar, Waqar Aqeel, Usman Nazir
LUMS School of Science and Engineering Pakistan



1 PROBLEM STATEMENT

Our main aim is to answer the question of verifiability i.e. the questions for example, whether the chosen route was the best one available, or whether it was consistent with the networks peering agreements, and privacy of crucial information (for example the route policies of the ASes) in SDXes. There are following three main components of our problem statement

A. Routing Problem from the characteristics of BGP

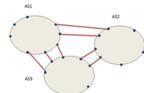


Fig-1 : Routing Problem

In BGP, Autonomous Systems (ASes) are abstracted as a node in a graph as shown due to privacy of intra-domain information, e.g., link quality, routing, flow info, policies etc. The problem with this approach is that traffic engineering by one AS can send flows over bad paths in neighboring ASs.

B. Inter-domain Routing Problem

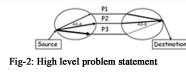


Fig-2: High level problem statement

As shown in figure 2, In Autonomous System (AS) A, the path associated with peering point P3 has most available bandwidth and choosing

this path will have direct impact of overloading on the link in AS-B. We have to design a technique so that neighboring domains conduct traffic engineering cooperatively in a scalable fashion without having to reveal confidential intra-domain information? Preventing this from happening requires that, sl, the available bandwidth on any link in B after traffic engineering by A, should be non-negative.

C. Information leakage problem

The second major problem is we cannot believe on single entity SDX which has all the (export) policies information made by autonomous systems. To get rid of these problems we propose our solution below, which requires no extra hardware or software.

4 RELATED WORK

The work most related to ours is [2], which used SMPC (Secure Multi-Party Computation) to solve the privacy preservation problem in an IXP setting. Detailed discussion on SDX and secure internet routing has been presented in many papers. There have been efforts in Verifying network-wide invariants in real time. Traffic engineering between neighboring domains is also done. A survey of interdomain routing policies is discussed. Some proposed private and verifiable interdomain routing decisions.

2 PROPOSED SOLUTION

For this purpose our global invariant is a triplet, which consist of encrypted load demands requests, available bandwidths from all ASes and function which provides above guarantees using one of our proposed model, given by the equation 1:

$$G = (\delta_{i,j}, [a_l, b_l, \dots], f()) \quad (1)$$

As shown in figure 4, we consider the case of two ASes with SDX. AS1 uses homomorphic public-key encryption and provides its public key to SDX and AS2 using its out-of-band relationship. The basic idea is SDX can verify the bandwidth using the equation 2

$$s_l = a_l - \sum \delta_{i,j} u_{i,j,l} \geq 0 \forall l \quad (2)$$

AS1 sends encrypted bandwidth $E(BW)$ and RPS to SDX to verify ABW. AS2 sends the encrypted load demand δ to SDX to establish link with AS1. The protocol is as follows:
 1) AS1 sends available bandwidths (ABWs) $E_A(a_l)$ to SDX.
 2) AS2 sends load demand $E_A(-\delta)$ to SDX.
 3) SDX compute the products (RBWs) $E_A(s_l)$ for various values of $E_A(a_l)$ by the homomorphic property.
 4) SDX then checks the value that belong to the RPS and selects that link for B to send traffic.
 5) The RBWs can also be used to serve other load demands requests.

In this whole process SDX doesn't know the load demand, ABWs and RBWs.

The proposed solution provides the following guarantees : (1) SDX does not know about policies information made by different ASes. (2) ASes will conduct traffic engineering in a scalable fashion without having to reveal confidential intradomain information. (3) Proof of concept protocols using homomorphic encryption and PSI

B. Proposed Model Using Private Set Intersection (PSI)

In this model, SDX will use the PSI based on Oblivious Transfer (OT) protocol [11] and selects the AS1 link which is suitable for AS2 load demand request. The protocol [11] is as follows:

- 1) AS1 and AS2 jointly choose an RSA modulus $n = pq$, such that neither knows its factorization. Let $\phi()$ be Euler's totient function. They can do so using any generic secure computation.
- 2) Furthermore using this secure protocol they generate the following exponents: d_1, d_2, e , and f . The exponents are generated, such that $(d_1 + d_2)e = f(\text{mod } \phi(n))$. These exponents are distributed as follows: SDX obtains e ; AS1 obtains d_1 and f ; AS2 obtains d_2 and f .
- 3) AS1 submits $\vec{x}' = x_1^{d_1}, \dots, x_v^{d_1} (\text{mod } n)$. AS2 submits $\vec{y}' = y_1^{d_2}, \dots, y_u^{d_2} (\text{mod } n)$.
- 4) SDX computes the $v \times w$ cross-product $\vec{z}' = \vec{x}' \times \vec{y}' = (x_1' y_1')^e, \dots, (x_v' y_w')^e$.
- 5) AS1 computes $x'' = x_1^f, \dots, x_v^f (\text{mod } n)$. AS1 and SDX engage in a regular, private set intersection protocol with the sets $x'' \cap \vec{z}'$. AS1 adds the respective element from \vec{z}' to the result set. AS2 performs the corresponding operations using his set \vec{y}' .

TABLE II. NOTATIONS USED FOR TEST SCRIPT

Variable	Description	Private to
$d_i, 1 \leq i \leq D$	Unique Destination Prefixes	None
$p_j, 1 \leq j \leq P$	Peering points between AS1 and AS2	None
$a_l; \vec{a} = (a_1, \dots, a_N)$	Links in AS1	AS1
$u_{i,j,l}; \vec{U}_{i,j} = (u_{i,j,l})_l$	1 if route to dest. d_i from p_j uses link a_l , 0 otherwise	AS1
$s_l, 1 \leq l \leq N$	Remaining bandwidth on link a_l of AS1	AS1
$\delta_{i,j}; \vec{\delta} = (\delta_{i,j})_{i,j}$	Load demand from destination d_i at p_j	AS1, AS2

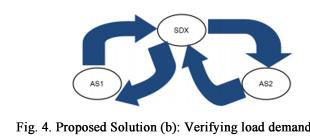


Fig. 4. Proposed Solution (b): Verifying load demand

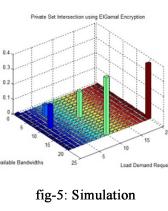


Fig. 5: Simulation

3 EVALUATION AND FUTURE WORK:

PSI execution times are shown in figure 5. Proposed model using homomorphic encryption takes $O(1)$ and proposed model using PSI takes $O(n)$ time. In the future we are trying to extend this concept over all the wide-area-networks and the data centers where this kind of communication and traffic engineering takes place. We are also working on making RSEs more secure rather than controller of SDX.

Fig. 3. Proposed solution (a): No need to hide internal links