Security Mechanisms for Traffic Engineering in SDX

Usman Nazir
LUMS School of Science and Engineering Pakistan

***Abstract*—Original architects of Inter-domain Internet Routing and Software Defined Exchange (SDX) paid less attention to accountability and security. This paper proposes a new inter-domain privacy-preserving security mechanisms with homomorphic encryption and private set intersection in complex networks with SDX.**

I. Introduction

Routing in BGP [1] depends only on destination IP prefix: BGP selects and exports routes for destination prefixes. Networks cannot make more finegrained decisions based on the type of application or the sender. BGP has Influence only over direct neighbors: Networks cannot directly express preferred inbound and outbound paths. Using SDX for inter-domain routing we get major benefits [2] as compared to BGP.
However, the main aim of this paper is to answer the question of verifability and privacy of routing decisions in SDX? Existing secure interdomain routing protocols can verify validity properties about individual routes, such as whether they correspond to a real network path [3]. It is often useful to verify more complex properties relating to the route decision procedure for example, whether the chosen route was the best one available, or whether it was consistent with the networks peering agreements. However, this is difficult to do without knowing a networks routing policy and full routing state, which are not normally disclosed. In this paper, we are trying to propose solution to how the centrlized controller (3rd party Route server) can allow the ASs connected to it to verify a number of nontrivial properties about inter-domain routing decisions without revealing any additional 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.

The remainder of the paper is organized as follows. Section II explains related work. SDX use-cases have been presented in Section III. Problem statement and proposed models have been elaborated in section IV and V respectively.

II. Related Work

Detailed discussion on SDX and secure internet routing has been presented in [1][2]. [3] proposed a routing security with privacy protections. [4] presented verifying global invariants in multi-provider distributed systems. Verifying network-wide invariants in real time is presented in [5]. Traffic engineering between neighboring domains is done in [6]. A survey of interdomain routing policies is discussed in [7]. [8] proposed private and verifiable interdomain routing decisions. Table I shows the classification of proposals for executing queries over encrypted data.

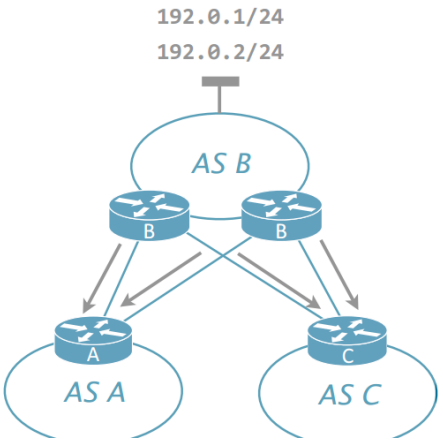

Fig. 1. Traffic Engineering

.

III. SDX use-cases with inter-domain routing Verifiability

A. Preventing Free-Riding

Today, an IXP member can easily free-ride by directing traffic to another member that did not export a corresponding BGP route (e.g., a peer dumping traffic destined to a member's other peer). An IXP could easily block such traffic by installing a 'drop' rule based on the input port, destination MAC address, and destination IP prefix, if the switches could support such rules, which SDX allows us to do. Since, the concerned Autonomous system (Call it C) will be depending on the controller to install this policy on the switch, it needs to verify that such a drop policy is actually in place. Controller (which is a third party) must not be able to fool the effected AS. So, we need to work on a mechanism via which C can verify only its concerned policies at the switch via the controller. Privacy of rest of the policies concerning other ASes must remain intact, and should not be revealed to C.

B. Inbound Traffic Engineering

An IXP member may want to divide incoming traffic over multiple router ports, based on the sender, the peer, or the application. Destination networks that do not even connect to the IXP could perform in-bound traffic engineering if the switches could encapsulate each packet with the address of one of its homing locations. SDX policies give any participant the

TABLE I. CLASSIFICATIONS OF PROPOSALS FOR EXECUTING QUERIES OVER ENCRYPTED DATA

PROPOSALS	PSIBased/PSIFree	HomomorphicBased/HomomorphicFree	Different Approach
Florian Kerschbaum [9]	PSIBased	HomomorphicBased	No
P Grofig, et. al. [10]	PSIFree	HomomorphicFree	SEEED
S. Machiraju, et. al. [4]	PSIFree	HomomorphicBased	No
Florian Kerschbaum [11]	PSIBased	HomomorphicFree	No
M. Beck, et. al. [12]	PSIFree	HomomorphicFree	String Matching
P Grofig, et. al. [13]	PSIFree	HomomorphicBased	Adjustable Encryption

direct control on its forwarding paths. For example, Bs SDX policy: B is an AS which has two BGP routers (L and R) to handle inbound traffic as shown in figure 1:
$match(dstip = X.X.X/24, srcmac = A), fwd(L))$
$match(dstip = X.X.X/24, srcmac = C), fwd(R))$
$match(dstip = X.X.X/24, srcmac = D), fwd(R))$
The policies mentioned above are engineering traffic coming from A to Bs L, and traffic coming from C and Ds to R. Now, though these are intra-domain related policies SDX provides us with the facility to bridge inter and intra domain policies at the switch. Now, again since its job of the Controller (which is not a trusted 3rd party) we need to verify that inbound intra policies are installed on the switch. Having, said so the issue in this case is that other ASes connected to controller are only concerned with sending packets to B. Hence, they are least interested to be concerned Bs inbound traffic engineering. Using, the work done in veriflow by obtaining a picture of the network as it evolves by sitting as a layer between the SDN controller and the forwarding SDX switch. Using the verify results B can ensure that Controller did actually implement the inbound policies. However, we have to modify veriflows work to make it work with POX controller and SDX forwarding fabric.

C. Traffic Offloading

Suppose member A has a peering relationship with member B but not member C. If A selects a BGP route with a path that traverses B followed by C, the traffic would flow through the IXP (and Bs link to the IXP) twice in traveling from A to B and ultimately to C. Forwarding the traffic directly from A to C would be more efficient, though ideally B could still charge A for the traffic that impinges on its relationship with C. Todays IXPs have limited control-plane mechanisms (e.g., third-party next-hop) for traffic offloading, but not for monitoring the offloaded traffic. Similar techniques as mentioned in the above two scenarios can be used to help B to ensure that even though traffic is not passing via B, Bs still get an illusion of the amount of traffic that would have passed via B. We plan to work more on this in future.

IV. PROBLEM STATEMENT

A. Routing Problem from the characteristics of BGP

In BGP, Autonomous Systems (ASs) are abstracted as a node in a graph as shown in figure 2 due to confidentiality 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 as explained in following high level problem statement.

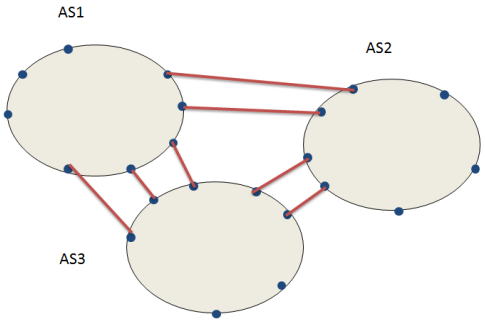

Fig. 2. Routing Problem

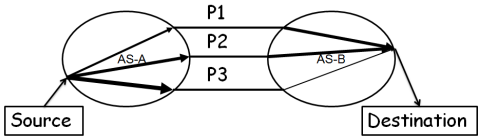

Fig. 3. High Level Problem Statement

B. Inter-domain Routing Problem

As shown in figure 3, In Autonomous System (AS) A, the path associated with peering point P3 has most available bandwidth path 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, s_l, 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 policies information made by autonomous systems. To get rid of these problems we proposed a solution below which requires no extra hardware or software.

V. PROPOSED MODELS

Proposed model provides following guarantees:

1) SDX does not know about policies information made by different ASs because all policies will be encrypted. (Addressing 'information leakage problem')
2) ASs will conduct traffic engineering in a scalable fashion without having to reveal confidential intra-domain information. (Addressing 'lack of knowledge about congestion in neighboring ASs' and 'bad intra-domain traffic engineering decision problem')
3) Proof of concept protocols using homomorphic encryption and PSI that verify global invariants ensuring

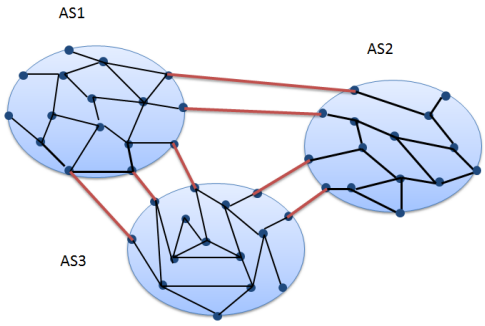

Fig. 4. Proposed solution (a): No need to hide internal links

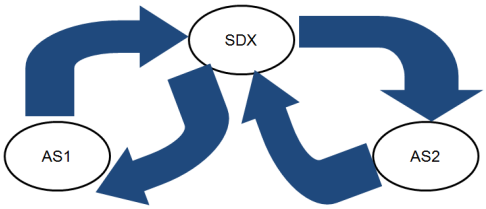

Fig. 5. Proposed solution (b): Verifying load demand

safe traffic engineering and verify policy safety in inter-domain routing.

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

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

A. Proposed Model Using Homomorphic Encryption

As shown in figure 5, we consider the case of two ASs 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 behind this protocol is that SDX can verify the membership of the remaining bandwidth (RBW) s_l as shown in equation 2 in the randomly permuted set (RPS) provided by AS1. The notations are explained in the table II

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

AS1 send 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 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.

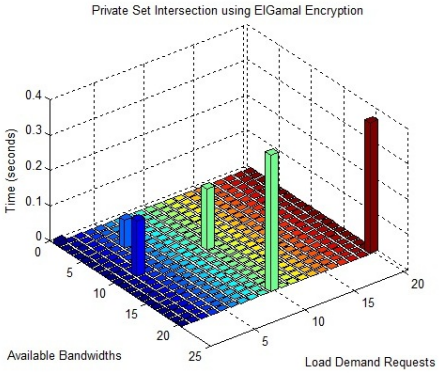

Fig. 6. Simulation

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: $d1, d2, e$, and f. The exponents are generated, such that $(d1 + d2)e = f(mod\phi(n))$. These exponents are distributed as follows: SDX obtains e; AS1 obtains $d1$ and f; AS2 obtains $d2$ and f.
3) AS1 submits $\overrightarrow{x}' = x_1^{d1}, ..., x_v^{d1}(modn)$. AS2 submits $\overrightarrow{y}' = y_1^{d2}, ..., y_w^{d2}(modn)$.
4) SDX computes the $v \times w$ cross-product $\overrightarrow{z} = \overrightarrow{x}' \times \overrightarrow{y}' = (x_1{}'y_1{}')^e, ..., (x_v{}'y_w{}')^e$.
5)AS1 computes $x'' = x_1^f, ..., x_v^f(modn)$. AS1 and SDX engage in a regular, private set intersection protocol with the sets x'' and $\overrightarrow{z}$, respectively. For each element in the intersection $x'' \bigcap \overrightarrow{z}$, AS1 adds the respective element from $\overrightarrow{x}$ to the result set. AS2 performs the corresponding operations using his set $\overrightarrow{y}$.

VI. Simulation Result

PSI execution times are shown in figure 6. Proposed model using homomorphic encryption takes O(1) and proposed model using PSI takes O(n) time.

References

[1] S. Goldberg, "Why is it taking so long to secure internet routing?" *Communications of the ACM*, vol. 57, no. 10, pp. 56–63, 2014.

[2] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett, "Sdx: A software defined internet exchange," in *Proceedings of the 2014 ACM conference on SIGCOMM*. ACM, 2014, pp. 551–562.

[3] A. J. Gurney, A. Haeberlen, W. Zhou, M. Sherr, and B. T. Loo, "Having your cake and eating it too: Routing security with privacy protections," in *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. ACM, 2011, p. 15.

[4] S. Machiraju and R. H. Katz, "Verifying global invariants in multi-provider distributed systems," in *Proc. SIGCOMM Workshop on Hot Topics in Networking (HotNets)*, 2004, pp. 149–154.

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
$a_l;\ \overrightarrow{A} = (a_1, ...a_N)$	Links in AS1	AS1
$u_{i,j,l};\ \overrightarrow{U}_{i,j} = (u_{i,j,l})_l$	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
$\delta_{i,j};\ \overrightarrow{\Delta} = (\delta_{i,j})_{i,j}$	Load demand from destination d_i at p_j	AS1, AS2

[5] A. Khurshid, W. Zhou, M. Caesar, and P. Godfrey, "Veriflow: verifying network-wide invariants in real time," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 467–472, 2012.

[6] J. Winick, S. Jamin, and J. Rexford, "Traffic engineering between neighboring domains," 2002.

[7] P. Gill, M. Schapira, and S. Goldberg, "A survey of interdomain routing policies," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 1, pp. 28–34, 2013.

[8] 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*. ACM, 2012, pp. 383–394.

[9] F. Kerschbaum, "Outsourced private set intersection using homomorphic encryption," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*. ACM, 2012, pp. 85–86.

[10] P. Grofig, M. Haerterich, I. Hang, F. Kerschbaum, M. Kohler, A. Schaad, A. Schroepfer, and W. Tighzert, "Experiences and observations on the industrial implementation of a system to search over outsourced encrypted data." in *Sicherheit*, 2014, pp. 115–125.

[11] F. Kerschbaum, "Collusion-resistant outsourcing of private set intersection," in *Proceedings of the 27th Annual ACM Symposium on Applied Computing*. ACM, 2012, pp. 1451–1456.

[12] M. Beck and F. Kerschbaum, "Approximate two-party privacy-preserving string matching with linear complexity," in *Big Data (BigData Congress), 2013 IEEE International Congress on*. IEEE, 2013, pp. 31–37.

[13] P. Grofig, I. Hang, M. Härterich, F. Kerschbaum, M. Kohler, A. Schaad, A. Schröpfer, and W. Tighzert, "Privacy by encrypted databases," in *Privacy Technologies and Policy*. Springer, 2014, pp. 56–69.