Picking a Partner: A Fair Blockchain Based Scoring Protocol for Autonomous Systems

Yousef Alowayed
KAUST

Marco Canini
KAUST

Pedro Marcos
UFRGS and FURG

Marco Chiesa
KTH Royal Institute of Technology

Marinho Barcellos
UFRGS

ABSTRACT

We tackle the problem of enabling Autonomous Systems to evaluate network providers on the basis of their adherence to Service Level Agreements (SLAs) regarding interconnection agreements. In current Internet practices, choices of interconnection partners are driven by factors such as word of mouth, personal relationships, brand recognition and market intelligence, and not by proofs of previous performance. Given that Internet eXchange Points provide increasingly more peering choices, rudimentary schemes for picking interconnection partners are not adequate anymore.

Although the current interconnection ecosystem is shrouded in confidentiality, our key observation is that recently-emerged blockchain technology and advances in cryptography enable a privacy-preserving decentralized solution based on actual performance measurements. We propose the concept of SLA score to evaluate network providers and introduce a privacy-preserving protocol that allows networks to compute and verify SLA scores.

CCS CONCEPTS

• **Networks** → **Network management**;

KEYWORDS

SLA score, AS ranking, blockchain

ACM Reference Format:
Yousef Alowayed, Marco Canini, Pedro Marcos, Marco Chiesa, and Marinho Barcellos. 2018. Picking a Partner: A Fair Blockchain Based Scoring Protocol for Autonomous Systems. In *ANRW '18: Applied Networking Research Workshop, July 16, 2018, Montreal, QC, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3232755.3232785

1 INTRODUCTION

The interconnection ecosystem is at the core of the Internet: it is a complex system that holds together hundreds of networks, providing the basic function of reaching anywhere from everywhere. Connections between networks – whether they are transit, or settlement-free peering or paid peering *agreements* – are strongly driven by business interests since most of the Autonomous Systems (ASes) are commercial. As such, ASes like network or service providers (and virtually any enterprise dependent on the quality of their Internet connectivity for business operations) must make informed decisions when planning with whom to interconnect [6, 14, 20]. Interconnection directly influences how traffic is routed across the Internet. Improving traffic delivery performance is believed to be critical for increasing customer satisfaction, brand image, and revenues, and for gaining a competitive advantage.

Finding and evaluating interconnection opportunities would require ASes to assess network providers regarding performance, reliability and ability to meet Service Level Agreements (SLAs). However, the current interconnection ecosystem is complex and shrouded in confidentiality: formal contracts are negotiated on a bilateral basis, agreements are typically regarded as confidential and proprietary information, and personal relationships and brand recognition carry significant influence [11, 13, 18]. Additionally, traffic data within these agreements is fundamentally considered private and jealously guarded. To gain more insights into the interconnection ecosystem, companies such as Telegeography and Dyn offer market data, statistics and AS rankings inferred through public routing information and external measurements. Yet, these approaches cannot determine the SLAs between two networks and, consequently, their ability to respect them.

The lack of a proper way to assess the performance of potential partners can result in missed interconnection opportunities. We posit that being able to evaluate potential providers regarding SLA adherence can help achieve better agreement decisions, and is necessary to fully leverage the

ANRW '18, July 16, 2018, Montreal, QC, Canada

ACM ISBN 978-1-4503-5585-8/18/07…$15.00
https://doi.org/10.1145/3232755.3232785

rich connectivity provided by Internet eXchange Points (IXPs) and colocation facilities (colos) [5].

We propose the use of *SLA scores* for helping ASes choose their business partners. We introduce a fair scoring protocol that allows the scores to be deterministically computed from measurements of forwarding performance. The proposed protocol provides the following guarantees: (i) ASes can deterministically compute and verify how scores were built; (ii) no benefit can be gained in delaying or hiding measurements; (iii) sensitive information about the interconnection agreements remains confidential; (iv) any AS attempting to modify its score will be detected; (v) false testimonies about the forwarding performance of another AS are detectable.

Supporting the above requirements is challenging due to inherently contrasting goals, i.e., third-party verification of the scores while preserving the privacy of the SLAs and traffic measurements of an agreement. Our contribution is the identification and careful combination of a set of key building blocks for our protocol: network measurements, order-preserving encryption [8], and a blockchain with smart contracts. Network measurements can be used to verify whether the forwarding performance is in conformance with SLAs of interconnection agreements. They are verified by smart contracts to compute an SLA score for each AS and to identify if any of the ASes has provided false testimony about forwarding performance. Order-preserving encryption enables any AS to compute and verify the scores without having access to the actual SLAs nor the measured values. Finally, the blockchain provides a persistent tamper-proof ledger, allowing ASes to check the scores and preventing undetected modifications. We note that alternative realizations of our concept are likely possible (e.g., by involving a trusted third-party); however, the properties of order-preserving encryption and the blockchain make them a good match for our requirements.

2 OVERVIEW

2.1 Subjective and Objective SLA scores

Consider an operator of an ISP that is connected to a large IXP. The operator wants to establish a new connectivity agreement that would allow them to reach a remote user base. Given the rich path diversity at IXPs, many networks will likely provide connectivity towards this user base, each with a different level of service (e.g., bandwidth, latency). The operator wants to make an informed decision among these possible provider alternatives: selecting a provider that fails to guarantee the agreed level of service would result in undesirable economic and reputation losses. One way to find a reliable provider is gathering informal word-of-mouth information from the operators' community. Yet, gathering this information is a process that takes non-negligible time and effort as it entails meeting other operators as well as convincing them to disclose information. Even worse, operators cannot distinguish whether an information is genuine or not. Ultimately, scoring a provider based on its perceived level of performance is *subjective* and cannot be *trusted*.

We argue that a better approach would be to characterize the quality of a provider according to trustworthy measurements of how well a provider has been able to meet its SLAs. We introduce the concept of an SLA score, an *objective* metric based on data that can be validated. The scores are objective and better enable an operator to select a provider.

2.2 Challenges of Objective SLA Scores

There are two major technical requirements for our vision. First, operators must be able to access the score of each provider. We satisfy it using a shared, tamper-proof ledger that stores this information. Second, the score must be verifiable. We rely on trusted measurements and a verifiable scoring protocol to update the scoring information.

Building an objective SLA score is challenging as it involves conciliating the contrasting *verification* goals with *privacy* requirements. Information about SLAs is confidential and must be kept private. Specifically, one has to guarantee that the SLA score written on the ledger solely depends on the measurement and SLA data of each agreement without revealing these data to other networks. Thus, a protocol implementing the concept of SLA scores needs to preserve the privacy of measurements and SLAs. It should output scores that are verifiably linked to said measurements or else, it must be possible to detect that the output is invalid. We now formalize the problem statement and then discuss the threat model and assumptions for our protocol.

2.3 Formal Problem Statement

We denote by s_a the SLA score of an agreement a. This is computed using any suitable function that depends on the measurement data and the agreed SLA for that agreement. We denote by s_n the SLA score of a network n. This is computed using any suitable function that depends on the set of (past) per-agreement scores involving n as a provider. We refer to this SLA score s_n as the *objective* score as it only depends on the measured data and the SLAs. We finally denote by l_n the SLA score written in the public ledger.

The goal of the SLA-SCORE problem is to design a protocol that guarantees, for each network n, that $l_n = s_n$ without having to disclose to any unintended party the per-agreement SLAs and its associated measurements.

2.4 Assumptions and Threat Model

Monitoring assumptions. We assume that, during an agreement, traffic data is collected by trusted hardware or a trusted third party. We refer to this party as the *Oracle*. We justify this assumption by observing that, since only two parties are involved in an agreement, it is impossible to achieve a fair exchange without trusted parties according to the impossibility results on fair exchange [19]. Thus, there must always be a neutral and trusted source of measurements, which may even be an intermediate IXP. We, however, attempt to limit their involvement as much as possible so that our protocol runs efficiently in the common case, and involves a more expensive verification phase only in case of disputes.

We further assume that measurements are digests and small enough to be stored on a blockchain efficiently without burdening a node’s storage or the network’s capacity. Previous work on verifiable network measurements also assumes small digests based on packet sampling [6].

Malicious attackers. We assume that any party involved in an agreement may misbehave to gain benefits. Specifically, a party may attempt to inflate its SLA score by altering the measured levels of performance or writing wrong SLA scores on the public ledger. Other parties not involved in the agreement may also misbehave with the goal of learning third-party SLAs or reducing the SLA score of other entities.

3 PROTOCOL

Using our protocol, honest parties will always receive an objective, measurement-based SLA score, while any cheating party will be detected and consequently penalized with the lowest possible score. To keep the scores of each interconnection agreement, we rely on a permissioned blockchain. A blockchain is a tamper-proof distributed ledger consisting of a growing number of blocks securely chained together, each block comprising of several records or transactions and a hash of the content of the previous block. Permissioned blockchains are resource-efficient and easy to maintain or upgrade, as they avoid the need for resources spent on achieving consensus, by limiting the numerous untrusted entities that can write to the blockchain. We rely on the IXP or colo to authorize parties on the blockchain. The data stored on the blockchain is manipulated using smart contracts, which are self-enforcing codes that enforce the execution of instructions.

For clarity, we describe the protocol regarding two network operators (participants) that we name customer (C) and provider (P). We assume a secure communication channel between C and P. The protocol comprises of three core phases: *setup*, *commit* and *open*, and two optional phases, invoked in the case of disputes: *verify* and *oracle invocation*. Through the *setup* phase, the participants store in the blockchain their encrypted commitments to exchange traffic with SLAs. Through the *commit* and *open* phases, the participants compute their SLA scores based on the agreed SLAs and the level of performance measured during the agreement. Through the *verify* and *dispute* phases, any party can verify the correctness of the computed scores via order-preserving encryption and smart contracts. To ensure termination, all but the first phase must be completed within a timeout duration that begins once an agreement concludes (more details in §3.7).

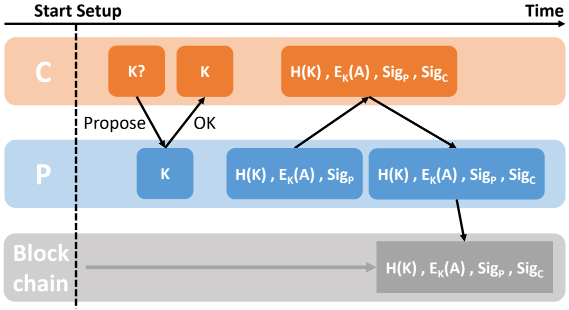
Start Setup
Time
C
K?
K
H(K) , E_K(A) , Sig_P , Sig_C
Propose
OK
P
K
H(K) , E_K(A) , Sig_P
H(K) , E_K(A) , Sig_P , Sig_C
Block chain
H(K) , E_K(A) , Sig_P , Sig_C

Figure 1: During the setup phase, a customer C and provider P reach an agreement on a set of SLA values A. They agree on a shared key K. P creates a setup block, and both P and C sign it before P publishes it to the blockchain

3.1 Setup

This initial setup phase of the protocol serves to publish a *setup block* signed by both C and P that commits them to an agreed SLA, a shared key, and to complete the protocol. The steps involved in this phase explained below, can be seen in Figure 1. Before exchanging traffic, both C and P stipulate the terms of the agreement. In most cases, this entails detailing the SLA, which possibly includes requirements for bandwidth, latency, packet loss, etc. We denote by A the formal description of the agreed SLA.

Next, the participants establish a shared secret key. C proposes a key K, which P saves and acknowledges. The provider then encrypts A, denoted with $E_K(A)$. The provider then creates a setup block, which includes $E_K(A)$ and a hash of the key $H(K)$. Finally, P signs the setup block with its private key, obtaining Sig_P and sends it to C, who checks the contents of the block as well as the signature. The customer then signs the block with its private key, obtaining Sig_C and sends it back to P. The provider checks the contents and signature and publishes the block onto the blockchain. From this moment, the block cannot be tampered with.

3.2 Commit

The commit phase (and the following open phase) follows a similar approach to Blum’s seminal coin flipping paper [7], which allows two parties to commit to their values before revealing them, thus ensuring fairness in this protocol.

At the conclusion of an agreement, the customer and the provider receive the same trace of network measurements, M_C and M_P, resp., from a trusted oracle O. Each party encrypts the received measurement using K, each producing an encrypted measurement $E_K(M_C)$ or $E_K(M_P)$. Assuming the encryption scheme is deterministic, the two encrypted measurements are identical, $E_K(M_C) = E_K(M_P)$, unless one of the party is

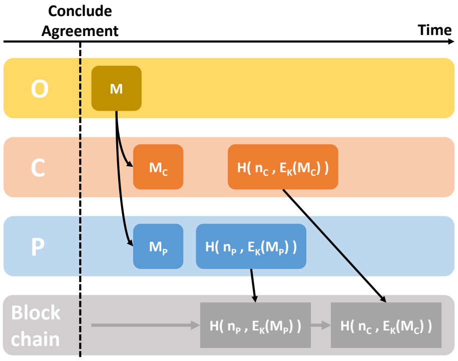

Figure 2: During the commit phase, C and P query network measurements from the oracle O after their agreement concludes. C and P encrypt their measurements and choose a random nonce n, then publish a commit block.

dishonest. Then, each participant generates a random nonce n that is used to create a unique hash. Next, the participants hash their nonce and encrypted measurements, $H(n, E_K(M))$ and add it to the blockchain, as shown in Figure 2. Nonces are used to avoid leaking information, which would give an advantage to the participant going second (see §4).

3.3 Open

Once both parties have committed to their measurements, they independently compute the scores using a predetermined scoring function $f_s(A, M)$, and then publish the scores on the blockchain. The scoring function must be pre-agreed upon and may be included in the original setup block or known publicly within the system.

If both scores are equal, the protocol terminates, and other nodes can now attribute these scores to the respective participants in the agreement. Otherwise, i.e., if the scores differ, the participants have a chance to defend their scores through the next phase of the protocol, the verify phase. If neither invokes the verify phase before the timeout, both are penalized for not completing the protocol, and other nodes can determine this non-completion by monitoring the blockchain.

3.4 Verify

The verify phase must retain the privacy of both the measurements and the SLA values, but also allow external verification of the scores. Only C and P have access to the unencrypted M and A. The other parties in the system have a verification function f_v such that the following invariant holds:

$$f_s(A, M) = f_v(E_k(A), E_k(M))$$

We describe below (§3.6) a possible approach to implement this verification phase based on order-preserving encryption, which is used due to the comparative nature of network measurements, but other encryption systems with similar properties could be applied (e.g., secure data types [21]).

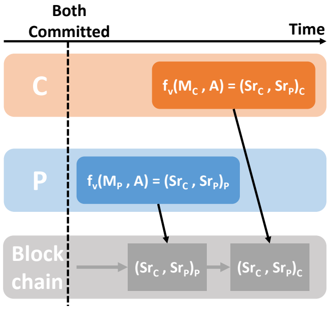

Figure 3: During the open phase, C and P publish the scores they computed to the blockchain.

One or both participants must publish a *verify block* to allow any other party, denoted N, to verify the validity of the scores published by the participant(s). The verify block includes the publisher's nonce and encrypted measurements. The case where only a single participant verifies the scores is simpler and will be discussed first.

Once a participant publishes a verify block, any other party can compute a score using both the verify and set up blocks as inputs. That party first checks that the nonce and encrypted measurements hash to the same value committed to by this participant. If it does not, this participant lied and is penalized, and the protocol terminates. If the hash values match, the verifying party uses the encrypted measurements from the verify block and encrypted SLA values from the setup block to compute the scores using f_v. If the output of f_v does not match the scores published by this participant, it lied; thus the participant is penalized, and the protocol terminates. But if the scores match, this participant computed the scores honestly (assuming the measurement has not been tampered). Now, the other participant has a chance also to publish a verify block. If the timeout expires, the other participant is deemed dishonest; it is penalized, and the protocol terminates.

In the case that the other participant also publishes a verify block, two cases are possible. First, if any of the participants lied, that participant is penalized, and the protocol terminates. Second, if neither participant lied about their scores, then one or both participants must have altered the measurements received from the oracle. In this case, the oracle must step in to remedy the situation of disputes.

While our current proposal requires a trusted oracle at this step, in our ongoing work we are exploring a version of the protocol based on zero-knowledge proofs [9]. This version will enable the participant to unequivocally prove that the scores are based on the trustworthy measurements without having to reveal anything about the SLAs nor measurements.

3.5 Oracle Invocation

If both participants were "truthful" in the previous phase, i.e., they have both correctly computed their scores based on the committed measurements, one or both of them must

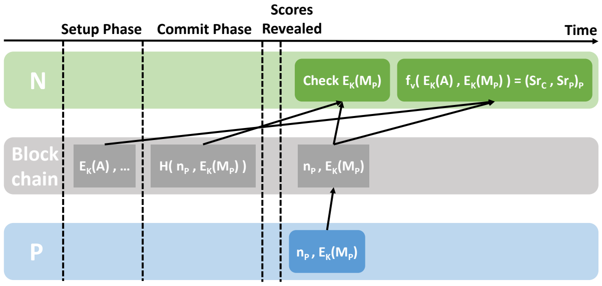

Figure 4: During the verify phase, P defends its computed score by publishing a verify block. This allows party N to check P’s commitment. Then N computes scores and checks them against the scores claimed by P.

have altered the measurements. This situation is reconciled by consulting the oracle that produced the measurements to determine which set of encrypted measurements, if any, is correct. At this point, the participants are given a chance to share the encryption key with the oracle. If neither of them shares the key, they are both penalized, and the protocol terminates. If one or both of them share the key, the oracle first checks if this key hashes to the key from the setup block. If it does not, the oracle waits for the correct key. If the key(s) matches, the oracle encrypts the correct measurements and publishes it on the blockchain.

Once the correct encrypted measurements are on the blockchain, any party can verify which participant was lying by checking the $E_K(M)$ from the verify blocks of the participants. The lying parties are penalized, and the protocol terminates.

3.6 Scores with Order Preserving Encryption

To preserve the privacy of the participants and allow other parties to verify any results, we use order-preserving encryption (OPE) [8]. The appealing property of OPE is that the ciphertexts preserve the ordering of the plaintexts: i.e., for any two values $x, y \mid x < y$, the relation $E(x) < E(y)$ also holds.

While our scheme is general, for simplicity, we now describe it assuming the SLA defines a constraint on the agreement’s bandwidth. Assume both parties agree on a bandwidth threshold $\bar{b}$ prior to exchanging traffic, and a set of timestamped measurements are taken periodically during the exchange: $M = \{(t_1, b_1), ..., (t_n, b_n)\}$. The scoring function could be defined as follows:

$$f_s(M, \bar{b}) = \frac{\text{COUNTIF}(b_i \in M \text{ if } b_i > \bar{b})}{\text{COUNT}(b_i \in M)}$$

That is, f_s returns the proportion of bandwidth measurements that adhere to the SLA threshold. If we encrypt both M and $\bar{b}$ using order preserving encryption, the following holds:

$$f_s(M, \bar{b}) = f_s(E(M), E(\bar{b}))$$

Thus, in this scenario, $f_s = f_v$ and this allows any party to compute the same score without knowing the original measurements M or bandwidth threshold $\bar{b}$. Other verification functions could be used but would require an appropriately selected encryption system [21] to achieve similar properties.

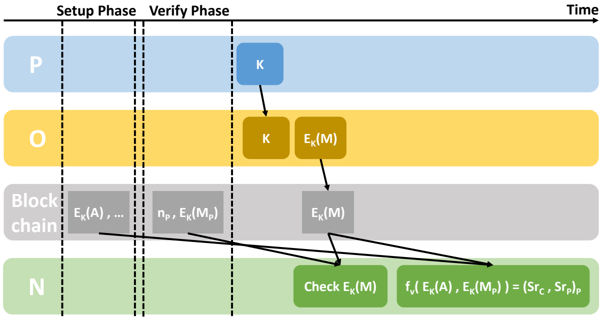

Figure 5: During the oracle invocation phase, P defends the validity of the measurements they used to compute their scores. P shares the key K with the oracle O. O publishes the encrypted measurements $E_K(M)$. Any party N can check these measurements and the scores computed from them against P’s claimed measurements and scores.

3.7 Additional Considerations

Timeouts. If the protocol does not terminate because one or both participants do not respond, the participant(s) responsible is penalized for promoting completion of the scoring process despite a potentially low score. Care must be taken in selecting the duration of the timeout to ensure that network latency or failures do not cause inadvertent termination of the protocol and the unfair punishment of honest participants.

Frequency. We advise that the scoring protocol be performed frequently enough so that (i) the scores reflect recent performance and (ii) libel attacks are deterred because they only impact a single score among many (see §4 for details).

Incremental deployment. The setup block is used to both reconcile any future scoring disputes by storing the agreed upon SLA values onto the blockchain. It also marks both parties’ commitment to continuing the protocol until termination. The deployment of the protocol involves the active participation of existing ASes which may not all join at once. Allowing ASes to opt in when desired avoids the problem of penalizing ASes that may be late in implementing the protocol. For our protocol to provide benefits, we require large adoption from many parties. We argue that adoption would follow if the benefits it provides would outweigh any of the deployment and operational complexity. These benefits serve as an incentive for adoption, as they establish a virtuous cycle.

4 SECURITY ANALYSIS

We demonstrate how this protocol deters and exposes dishonest parties while maintaining privacy in the face of attacks.

Fairness. Fairness is achieved through the commit and open phases because no information is leaked until both parties have committed to their measurements. A random nonce ensures that the participant to commit second cannot know whether their measurements match that of the other participant. Thus, committing or opening first or second does not give either participant additional knowledge or an advantage over the other.

Penalties. Low scores are used as penalties to deter and penalize dishonest behavior. Honest parties are given the score they propose. An honest party could also be given a null score if the other is caught cheating, depending on the desired application.

Do Nothing Attack. A participant is penalized when they sign and publish a valid setup block but then do not publish the commit, open and verify blocks in their respective phases before the timeout elapses. This protects honest parties that are guaranteed to receive their scores before the timeout.

Partition and Eclipse Attack. An adversarial participant may intercept and block network traffic to stop the other party from publishing blocks until the timeout elapses. Although unlikely, a powerful adversarial party may force an honest participant to be penalized for not completing the protocol. Eclipse attacks have been shown to be plausible in Bitcoin but require a large number of bots [16]. In our case, having a sufficiently long timeout will drastically decrease the impact of eclipse attacks since at least one other party will at some point receive blocks from the attacked honest party.

Lucky Attack. A dishonest party may receive a false score because the other party was caught cheating. Assume a party X commits to altered measurements. If scores do not match in the open phase, each participant must publish the measurements they committed to. If the other party Y publishes measurements that do not match their commitment and X publishes measurements that match, Y is penalized, and X receives their self-assigned scores. Although X receives fabled scores, we believe this attack to be uncommon and difficult for an attacker to predict, because it is in Y's interest to also alter their measurements if they were to attempt to lie. Moreover, other parties that observe the protocol executions may ignore self-assigned scores.

Libel Attack. A dishonest node may cheat on purpose to force the honest node's score to be self-assigned. Other parties may disregard this self-assigned score since its validity was never disputed and there is a chance it was fabricated. Such an attack counters the attacker's best interest and results in them receiving a low score, which makes it unlikely. The attack can be mediated by increasing the frequency of the protocol, such that a single libel attack only affects one score among many, as advised in §3.7.

Kamikaze Attack. An adversarial party can push the protocol to the last phase, oracle invocation, by cheating. During oracle invocation, the honest party has to share the key K with the oracle, which allows the oracle to decrypt the original SLA values A. If the honest party does not want to disclose this information, it will subsequently be penalized for not completing the protocol. Ongoing work with zero-knowledge proofs [9] hopes to remedy this vulnerability.

Collusion Attack. The protocol cannot detect collusions between participants to produce matching scores that do not match the measurements. We believe this type of collusion will be uncommon because it implies a loss to customers who have not received the performance expected. Parties could pay to collude and increase their score, which would go undetected and should be a consideration in future iterations of this protocol.

5 RELATED WORK

The problem of forwarding performance verification has been studied before. Network Confessional [6] relies on voluntary reporting by ISPs and enables verifiable network-performance measurements. The broad problem of improving the interconnection ecosystem – an inherent tussle of economics [12] – has lately received attention from academia and industry. Prior academic works in this area recognize that there are missed interconnection opportunities and propose to address this problem by establishing marketplaces for connectivity, facilitating their negotiation processes [23, 24], and by automating the process of establishing interconnection agreements [10, 17]. At industry, several companies offer on-demand connectivity with cloud providers for networks already connected to their points of presence [1–4]. We view our work as orthogonal to these efforts in that existing works seek to facilitate interconnection whereas our approach facilitates evaluating potential providers *ex ante*.

The choice of network providers affects end users shopping for home broadband service plans. Sundaresan et al. [22] proposed to label each ISP service plan with comprehensive information about network metrics like throughput, latency, loss rate, and jitter. Although these metrics are also used for specifying SLAs, our context is very distinct from labeling broadband providers. Recently Hari and Lakshman [15] proposed the use of a blockchain based mechanism to secure the Internet BGP and DNS infrastructure. We share their view that blockchain technology fulfills the need for a tamper-resistant framework that is outside the control of any single entity and is useful for improving Internet operations.

6 SUMMARY

In this paper, we demonstrated the need for a trustworthy, verifiable and privacy conserving mechanism for scoring ASes. Then we presented a protocol that allows for the computation of these scores that is fair to both participants and has the ability to expose cheaters. Finally, we analyzed the security of this protocol in the face of several attacks and network issues. Implementing and evaluating this proposal is part of our ongoing work.

REFERENCES

[1] Console Connect – The Cloud Connection Company, 2018. https://www.consoleconnect.com/.

[2] Epsilon Telecommunications – Connectivity made simple, 2018. http://www.epsilontel.com.

[3] Megaport – A Better way to connect, 2018. http://megaport.com.

[4] Packetfabric, 2018. http://www.packetfabric.com.

[5] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger. Anatomy of a Large European IXP. In *SIGCOMM*, 2012.

[6] K. Argyraki, P. Maniatis, and A. Singla. Verifiable Network-Performance Measurements. In *CoNEXT*, 2010.

[7] M. Blum. Coin Flipping by Telephone a Protocol for Solving Impossible Problems. *ACM SIGACT News*, 15(1), 1983.

[8] A. Boldyreva, N. Chenette, Y. Lee, and A. O'Neill. Order-Preserving Symmetric Encryption. In *EUROCRYPT*, 2009.

[9] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell. Bulletproofs: Short Proofs for Confidential Transactions and More. *Cryptology ePrint Archive*, 2017/1066, 2017.

[10] I. Castro, A. Panda, B. Raghavan, S. Shenker, and S. Gorinsky. Route Bazaar: Automatic Interdomain Contract Negotiation. In *HotOS*, 2015.

[11] M. Chiesa, D. Demmler, M. Canini, M. Schapira, and T. Schneider. SIXPACK: Securing Internet eXchange Points Against Curious onlooKers. In *CoNEXT*, 2017.

[12] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden. Tussle in Cyberspace: Defining Tomorrow's Internet. *IEEE/ACM Trans. Netw.*, 13(3), June 2005.

[13] P. Faratin, D. D. Clark, S. Bauer, W. Lehr, P. W. Gilmore, and A. Berger. The Growing Complexity of Internet Interconnection. *Communications & Strategies*, 72(4th Quarter 2008), Dec. 2008.

[14] C. Hall, R. Clayton, R. Anderson, and E. Ouzounis. Inter-X: Resilience of the Internet Interconnection Ecosystem. Technical report, ENISA, Apr. 2011. Panagiotis Trimintzios, editor.

[15] A. Hari and T. V. Lakshman. The Internet Blockchain: A Distributed, Tamper-Resistant Transaction Framework for the Internet. In *HotNets*, 2016.

[16] E. Heilman, A. Kendler, A. Zohar, and S. Goldberg. Eclipse Attacks on Bitcoin's Peer-to-Peer Network. In *USENIX Security Symposium*, 2015.

[17] P. Marcos, A. Wermann, L. Bertholdo, and M. Barcellos. DYNAMIX - A Dynamic Agreement Markeplace on Internet eXchange Points. In *CoNEXT Student Workshop*, 2016.

[18] W. B. Norton. *The Internet peering playbook: connecting to the core of the Internet*. DrPeering Press, 2014.

[19] H. Pagnia and F. C. Gärtner. On the Impossibility of Fair Exchange without a Trusted Third Party. Technical Report TUD-BS-1999-02, Darmstadt University of Technology, 1999.

[20] A. Popescu, T. Underwood, and E. Zmijewski. Rankings, Damned Rankings & Statistics, Nov. 2007. Presentation at MENOG 2. Available at https://dyn.com/wp-content/uploads/2013/05/menog2.pdf.

[21] S. Savvides, J. J. Stephen, M. S. Ardekani, V. Sundaram, and P. Eugster. Secure Data Types: A Simple Abstraction for Confidentiality-preserving Data Analytics. In *SoCC*, 2017.

[22] S. Sundaresan, N. Feamster, R. Teixeira, A. Tang, W. K. Edwards, R. E. Grinter, M. Chetty, and W. de Donato. Helping Users Shop for ISPs with Internet Nutrition Labels. In *HomeNets*, 2011.

[23] V. Valancius, N. Feamster, R. Johari, and V. Vazirani. MINT: A Market for INternet Transit. In *ReArch*, 2008.

[24] T. Wolf, J. Griffioen, K. L. Calvert, R. Dutta, G. N. Rouskas, I. Baldin, and A. Nagurney. ChoiceNet: Toward an Economy Plane for the Internet. *SIGCOMM Comput. Commun. Rev.*, 44(3), 2014.