Aaron D. Jaggard*, Aaron Johnson, Sarah Cortes, Paul Syverson, and Joan Feigenbaum

20,000 In League Under the Sea: Anonymous Communication, Trust, MLATs, and Undersea Cables

Abstract: Motivated by the effectiveness of correlation attacks against Tor, the censorship arms race, and observations of malicious relays in Tor, we propose that Tor users capture their trust in network elements using probability distributions over the sets of elements observed by network adversaries. We present a modular system that allows users to efficiently and conveniently create such distributions and use them to improve their security. To illustrate this system, we present two novel types of adversaries. First, we study a powerful, pervasive adversary that can compromise an unknown number of Autonomous System organizations, Internet Exchange Point organizations, and Tor relay families. Second, we initiate the study of how an adversary might use Mutual Legal Assistance Treaties (MLATs) to enact surveillance. As part of this, we identify submarine cables as a potential subject of trust and incorporate data about these into our MLAT analysis by using them as a proxy for adversary power. Finally, we present preliminary experimental results that show the potential for our trust framework to be used by Tor clients and services to improve security.

Keywords: Tor, Trust, Bayesian Belief Network, MLAT, submarine cable

DOI DOI
Received ..; revised ..; accepted ...

***Corresponding Author: Aaron D. Jaggard:** U.S. Naval Research Laboratory, E-mail: aaron.jaggard@nrl.navy.mil
Aaron Johnson: U.S. Naval Research Laboratory, E-mail: aaron.m.johnson@nrl.navy.mil
Sarah Cortes: Northeastern University, E-mail: scortes@ccs.neu.edu
Paul Syverson: U.S. Naval Research Laboratory, E-mail: paul.syverson@nrl.navy.mil
Joan Feigenbaum: Yale University, E-mail: joan.feigenbaum@yale.edu

1 Introduction

Tor and its users currently face serious security risks from adversaries positioned to observe traffic into and out of the Tor network. Large-scale deanonymization has recently been shown feasible [22] for a patient adversary that controls some network infrastructure or Tor relays. Such adversaries are a real and growing threat, as demonstrated by the ongoing censorship arms race [12] and recent observations of malicious Tor relays [32]. In light of these and other threats, we propose an approach to representing and using trust in order to improve anonymous communication in Tor. Trust information can be used to inform path selection by Tor users and the location of services that will be accessed through Tor, in both cases strengthening the protection provided by Tor. A better understanding of trust-related issues will also inform the future evolution of Tor, both the protocol itself and its network infrastructure. Path selection and the evolution of the protocol and infrastructure will also be informed by a more comprehensive understanding of potential adversaries.

Attacks on Tor users and services include first–last correlation [28], in which an adversary correlates traffic patterns between the client and an *entry guard* (i.e., a relay used by a client to start all connections into Tor) with traffic patterns between a Tor *exit* (i.e., a relay that will initiate connections outside the Tor network) and a network destination in order to link the client to her destination. They also include more recently identified attacks on a single end of a path such as fingerprinting users [6] or attacking hidden services [4]. With trust information, users could choose trusted paths through the Tor network and services could choose server locations with trusted paths into the network in order to reduce the chance of these attacks.

We propose a modular system that (i) allows users to express beliefs about the structure and trustworthiness of the network, (ii) uses information about the network, modified according to the user-provided structural information, to produce a "world" that captures how compromise is propagated through the network,

and (iii) combines this world with the user's trust beliefs to produce a Bayesian Belief Network (BBN; see, e.g., [14]) representing a distribution on the sets of network elements that an adversary might compromise. The system we describe is designed to produce a distribution on the sets of network locations that might be compromised by a single adversary. In the case of multiple, non-colluding adversaries, multiple distributions could be produced.

We illustrate how this system might work by introducing two novel types of adversaries. First, we consider a powerful, pervasive adversary called *The Man* that is potentially observing any independent group of of Autonomous Systems (ASes), Internet Exchange Points (IXPs), or relay families. The user is uncertain about exactly what this adversary can observe, but she has some information about the risk at different locations. This adversary can be seen as a generalization of previous threat models in which an adversary might compromise relays in the same /16 subnet or family, or in which an individual AS or IXP might be malicious.

Second, we initiate the study of the effects of Mutual Legal Assistance Treaties (MLATs) on the reach of adversaries; we also identify submarine cables as potentially important subjects of (dis)trust and incorporate data about these into our analysis. Here, we demonstrate the use of an MLAT database to inform analysis of first–last compromise. The randomized state-level adversaries that we construct for this make use of data on submarine cables, opening up that avenue of study in connection with anonymity networks. We use existing Tor traceroute data to give an initial understanding of how MLATs may expand the capabilities of adversaries.

In addition, we present proof-of-concept experiments that show how our trust system might be used by client or servers to improve their security. We suppose that users choose paths and servers choose locations to minimize the risk of first–last correlation attacks by The Man. The results show that users and services can employ our system to improve their security.

The main part of our modular system was described in an unpublished paper [19]. The version presented here explicitly accounts for MLATs in the way that we use them, a modification that demonstrates the flexibility of our system. Our use of the MLAT and cable databases and our analysis of the effects of MLATs on the reach of adversaries are also new since that preliminary version of this work.

Other work [20, 21] has considered the use of trust to improve security in Tor. The models of trust in this previous work have the major limitations that they only can be used to describe Tor relays and that they assume each relay has an independent chance of compromise. The framework we present here represents a significant advance in that it includes a diverse set of network elements, including elements such as IP routers or IXPs that exist only on the paths *between* Tor relays. We allow new types of network elements to be added in natural ways. Another contribution of our system is that it can be used to represent arbitrary probability distributions over the sets of network elements, and yet we show how the most likely distributions can be efficiently represented and used.

The body of this paper provides a high-level view of our system, starting with an overview of its operation and what the system provides in Sections 2 and 3. We describe in Section 4 how the system-provided information is combined with user beliefs to produce a BBN. We discuss some issues related to users' trust beliefs in Section 5. We present The Man in Section 6. In Section 7, we discuss MLATs and analyze their implications for adversary capabilities; the randomized construction of hypothetical adversaries for that analysis is guided by countries' connections to submarine cables. We then present, in Section 8, experimental results from our trust system. We close in Section 9 with a discussion of the implications of the work presented here and a sketch of ongoing and future work. As noted throughout, some additional details are provided in the appendices.

2 System Overview

We survey our system, which is largely modular. This allows it to be extended as new types of trust information are identified as important, etc. The system comes with an ontology that describes types of network elements (e.g., AS, link, and relay-operator types), the relationships between them that capture the effects of compromise by an adversary, and attributes of these things. While we provide an ontology, this may be replaced by another ontology as other types of threats are identified. Section 3.1 describes the requirements for replacement ontologies. Roughly speaking, the ontology identifies the types of entities for which the system can automatically handle user beliefs when constructing the Bayesian Belief Network (BBN) for the user. A user may express beliefs about other types of entities, but she would need to provide additional information about how those entities relate to entities whose types are in the ontology.

The ontology is provided to the user in order to facilitate this.

In general, we expect that the system will provide information about network relationships, such as which ASes and IXPs are on a certain virtual link or which Tor relays are in a given relay family. We generally expect the user to provide information about human–network relationships such as which individual runs a particular relay. Note that this means the user might need to provide this type of information in order to make some of her beliefs usable. For example, if she has a belief about the trustworthiness of a relay operator, she would need to tell the system which relays that operator runs in order for the trustworthiness belief to be incorporated into the BBN.

Using the ontology and various published information about the network, the system creates a preliminary "world" populated by real-world instances of the ontology types (e.g., specific ASes and network operators). The world also includes relationship instances that reflect which particular type instances are related in ways suggested by the ontology. User-provided information may include revisions to this system-generated world, including the addition of types not included in the provided ontology and instances of both ontology-provided and user-added types. The user may also enrich the information about the effects of compromise (adding, e.g., budget constraints or some correlations).

The user expresses beliefs about the potential for compromise of various network entities; these beliefs may refer to specific network entities or to entities that satisfy some condition, even if the user may not be able to effectively determine which entities satisfy the condition. This user-provided information is used, together with the edited world, to create a Bayesian Belief Network (BBN) that encodes the probability distribution on the adversary's location that arises from the user's trust beliefs. A user may express a belief that refers to an entity or class of entities whose type is in the given ontology. For such beliefs, the system will be able to automatically incorporate those beliefs into the BBN that the system constructs. A user may also express beliefs about entities whose types are not included in the ontology. If she does so, she would need to provide the system with information about how those entities should be put into the BBN that the system constructs.

The system and the user need to agree on the language(s) in which she will express her beliefs. Different users (or, more likely, different organizations that want to provide collections of beliefs) may find different languages most natural for expressing beliefs. The language specification(s) must describe not only the syntax for the user but also (i) how her structural beliefs will be used in modifying the system-generated world and (ii) how her other beliefs will be used to translate the edited world into a BBN.

Once constructed, the BBN can be used, e.g., to provide samples from the distribution of the Tor relays and Tor "virtual links" (transport-layer connections with Tor relays) that are observed by the adversary. The motivating application is to use these samples to inform more secure path selection in Tor.

2.1 Construction sequence

An overview of the system's actions is as follows. The various attributes and beliefs mentioned here are described in detail in the following sections.

1. World generation from ontology: ${}_{R}W_{T}^{I}$
   - As described in Section 3.3, the system generates a preliminary view of the world based on the ontology and its data sources. We denote the result by ${}_{R}W_{T}^{I}$.
   - This includes system attributes.
2. Augmenting the types with the user's types: ${}_{R}W_{T'}^{I}$
   - The user may provide additional types (as a prelude to adding instances of those types to the world). We use ${}_{R}W_{T'}^{I}$ to denote the augmentation of ${}_{R}W_{T}^{I}$ by adding the user's types.
3. Adding user-specified instances of types (ontology and user-provided): ${}_{R}W_{T'}^{I'}$
   - The user may add instances of any of the types in ${}_{R}W_{T'}^{I}$. We use ${}_{R}W_{T'}^{I'}$ to denote the augmentation of ${}_{R}W_{T'}^{I}$ by adding these new instances and removing any that the user wishes to omit.
4. Adding user-specified relationships (between instances in ${}_{R}W_{T'}^{I'}$): ${}_{R'}W_{T'}^{I'}$
   - The user may specify additional parent/child relationships beyond those included in ${}_{R}W_{T'}^{I'}$. In particular, any new instances that she added in the previous step will not be related to any other instances in the world unless she explicitly adds such relationships in this step. We use ${}_{R'}W_{T'}^{I'}$ to denote the augmentation of ${}_{R}W_{T'}^{I'}$ by adding these new relationships and by removing any that the user wishes to omit.
5. Edit system-provided attributes (not budgets or compromise effectiveness).
6. Add new user-provided attributes.
7. Add budgets.

8. Add compromise effectiveness (if values are not given, this defaults to a value provided by the ontology; for relationships of types not given in the ontology, we will use a default value unless the user specifies something when providing the relationship instance).
9. Produce BBN.
   – In this overview, this process is treated as a black box. In practice, it involves many steps that depend on the belief language used. The procedure for the belief language described in Sec. 4.2 is presented in detail in Sec. 4.3.

3 Ontology and World

Before presenting the ontology that we use in this work, we describe our general requirements for ontologies in this framework. This allows our ontology to be replaced with an updated version satisfying these requirements.

3.1 General requirements for ontologies

We assume that any ontology used in our system has the following properties:

– It has a collection $\mathcal{T}$ of *types*. We use the ontology to describe relationships between the types in the ontology.
– A collection $\mathcal{E}$ of (directed) *edges* between types (with $\mathcal{E} \cap \mathcal{T} = \emptyset$). The edges are used to specify relationships; if there is an edge from T_1 to T_2 in the ontology, then the compromise of a network element of type T_1 has the potential to affect the compromise of a network element of type T_2.
– Viewed as a directed graph, $(\mathcal{T}, \mathcal{E})$ is a DAG.
– A distinguished set of $\mathcal{T}$ called the *output types*. This is for convenience; these are the types of instances that we expect will be sampled for further use. We generally expect the output types to be exactly the types in the ontology that have no outgoing edges.
– Each element of $\mathcal{T} \cup \mathcal{E}$ has a *label* that is either "system" or "user." For an edge e from type T_1 to type T_2, if either T_1 or T_2 has the label "user," then e must also have the label "user." These labels will be used to indicate the default source of instances of each type. (However, the user may always override system-provided information.)
  Types or edges with the label "user" might be natural to include in an ontology when the type/edge is something about which the system cannot reliably obtain information but the ontology designer is able to account for instances of the edge/type in the BBN-construction procedure.
– A collection $\mathcal{A}$ of *attributes*. Each attribute includes a name, a data type, a source (either "system" or "user"). Each element of $\mathcal{T} \cup \mathcal{E}$ may be assigned multiple boolean combinations of attributes; each combination is labeled with either "required" or "optional."1

Other ontologies may modularly replace the one described here if they satisfy the assumptions described above.

3.2 Our ontology

Figure 1 shows the elements of our ontology. Rounded rectangles are types; instances of these will be factor variables in the BBN produced by the system. Ovals are output types: Tor relays and (virtual) links between clients and guards and between exits and destinations. Cylinders are attributes, whose interpretation is described below. With the exception of Relay Software and Physical Location, which the system provides but the user may modify, these attributes are provided by the user. The user may also provide new attributes.

Directed edges show expected relationships between types. For example, the edge from the "AS" type to the "Router/switch" type indicates that we expect that the compromise of an AS will likely contribute to the compromise of one or more routers and switches. This edge is dashed in Fig. 1 to reflect the label "user," i.e., we currently expect the user to identify which AS controls a particular router or switch if that effect is to be incorporated into the BBN construction. Other dashed edges and the unfilled types/attributes are also elements that we expect to be provided by the user. Solid edges and filled-in types correspond to elements and attributes whose label is "system;" we expect the system provide information about these.

1 In the rest of this paper, we assume that each combination is just a single "optional" attribute without any connectives. The semantics of individual attributes depend on the translation procedure that produces the BBN. We expect that a boolean combination of attributes will be interpreted as possible combinations of attributes that the translation procedure can handle; for example, it might be able to process either a pair of integers or a single real value. Richer applications of the "optional" and "required" labels might be allowed as well, although we do not need them here.

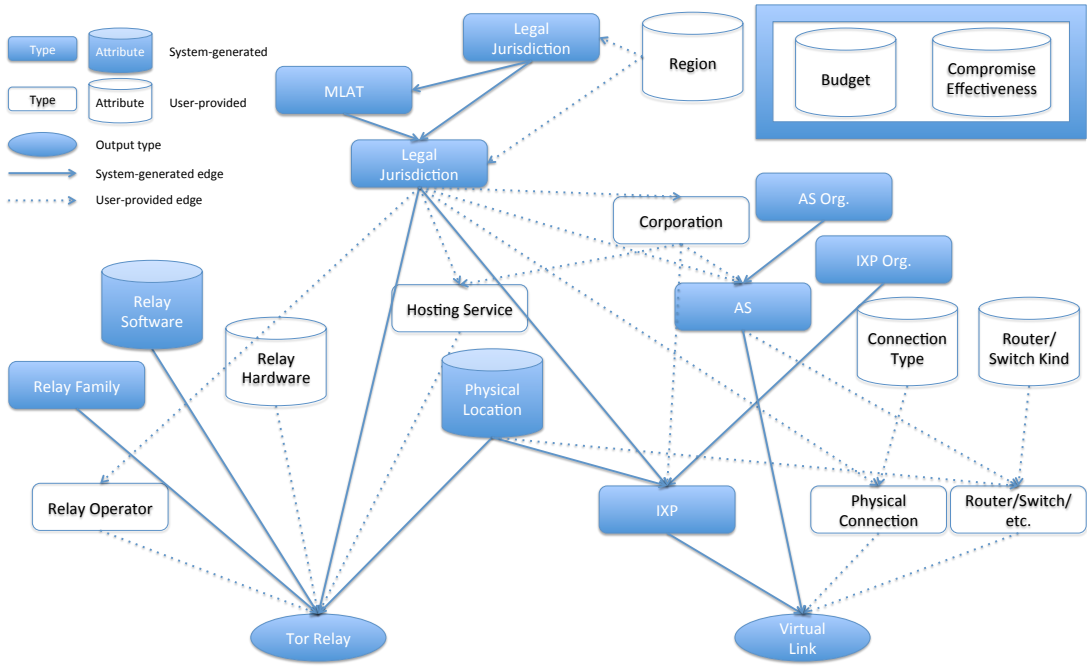

Fig. 1. Graphical depiction of the system's ontology

3.2.1 User-provided types

The types and relationships that are provided by the system in constructing the preliminary world are described in Section 3.3. We describe the others here; instances of these are added by the user in ways specified below.

Hosting Service (and incident edges) Hosting services that might be used to host Tor relays. If a service hosts a particular relay, there would be a relationship instance from the service to the relay. If a service is known to be under control of a particular legal jurisdiction or company, the appropriate incoming relationship instance can be added.

Corporation (and incident edges) Corporate control of various network elements may be known. A corporation that is known may be added as an instance of this type. If the corporation is known to be subject to a particular legal jurisdiction, then a relationship edge from that jurisdiction to the corporation can be added. Similarly, hosting services, ASes, and IXPs that a corporation controls may be so indicate via the appropriate relationship instances.

Router/switch/etc. This corresponds to a physical router or switch. We do not attempt to identify these automatically, but ones known to the user (or a source to which the user has access) may be added as instances of this type.

Physical connection Particular physical connections, such as a specific cable or wireless link, may be known and of interest.

(Physical connection, Virtual link) If a virtual link is known to use a specific physical connection, then that can be reflected in a relationship between the two.

3.2.2 Attributes

The attributes in our ontology are depicted by cylinders in Fig. 1. The two at the box in the top right can be applied to all non-output type instances, so we do not explicitly show all of the types to which they can be applied.

System-generated attributes These include relay-software type and router/switch type. Users may edit these, e.g., to provide additional information.

Connection type This is an attribute of physical-connection instances. It is represented as a string that describes the type of connection (e.g., `"submarine cable"`, `"buried cable"`, or `"wireless connection"`). A user would express beliefs about connection types; if the type of a connection is cov-

ered by the user's beliefs, then the probability of compromise would be affected in a way determined by the belief in question.

Budget This attribute, which is supplied by the user at her option, may be applied to any non-output type instance. There are two variants. Both are represented as an integer k and another value. In the first variant, the other value is a type; in the second variant, the other value is the string `"all"`. Multiple instances of this attribute may be applied to a single type instance as long as they have distinct second values; if one of these is the second variant, then all others will be ignored. This allows the user to express the belief that, if the type instance is compromised, then its resources allow it to compromise k of its children. In the first variant of this attribute, the instance may compromise k of its children of the specified type (and perhaps k' of its children of a different type, if so specified by a different belief). In the second variant of this attribute, the instance may compromise k of its children across *all* types.2

As discussed below, we must approximate the effects of resource constraints so that the BBN can be efficiently sampled.

Region This is an attribute of legal jurisdiction. It is represented as a boolean predicate on geographic coordinates.

Compromise effectiveness This attribute is syntactically similar to the budget attribute. It is supplied by the user at her option for instances of any non-output type, and there are effectively two variants. This is represented as a probability $p \in [0, 1]$ and a boolean predicate on type instances; we distinguish non-trivial predicates from the always-true predicate $\top$. Multiple instances of this attribute may be applied to a single type instance as long as no two non-$\top$ predicates evaluate to True on the same input. Only one instance of this attribute with $\top$ may be present; if it is, then all other instances of the attribute for the type instance are ignored.

This attribute allows the user to express beliefs about the effect of compromise of one type instance on its children, either uniformly or according to type. For example, a compromised AS might attempt to compromise all of its routers; with some probability (e.g., $p = 10^{-4}$), it might make a mistake in the configuration file for a certain router model that would prevent it from compromising routers of that model that are not otherwise compromised. However, if such a mistake is not made, then the AS will compromise *all* routers of that model; this is in contrast to the effects of budget beliefs.

Router/Switch Kind This is an attribute of routers/switches and is represented as a set of strings. We expect the user to use this to describe aspects of routers/switches that she might know about and want to use in her trust beliefs, e.g., the model number or firmware version of specific routers and switches.

Relay Hardware This is an attribute of relays and is represented in the same way as the router/switch kind. Also analogously to that attribute, we expect that the user would use this to describe aspects of relay hardware that she might know about and potentially use in her trust beliefs.

3.3 System-generated world

The system provides users with a *world* consisting of *type instances* and *relationship instances* that are consistent with the types and relationships specified in the ontology. Formally, a world is a DAG in which each vertex is a type instance, each edge is a relationship instance, and an attribute function assigns each vertex a vector of attributes. A type instance represents a real-world object of the specified type. For example, "AS3356" is a type instance of the AS type, and "Level 3 Communications" is a type instance of the AS Organization type. A relationship instance will only relate two instances of types that are related in the ontology. For example, (Level 3 Communications, AS3356) is an instance of the (AS Organization, AS) relationship type and indicates that AS3356 is a member of Level 3 Communications. The attributes of a type instance provide information that users can incorporate into their trust beliefs, such as the location of a given Tor relay. The world can be modified by users in ways provided by the trust language. We assume that each instance has a unique identifier and an indication of the type of which it is an instance.

For our ontology, the system generates a world as follows:

1. The current Tor consensus and the server descriptors it references are used to create the following instances and attributes, which concern relays:

2 The resources needed to compromise instances of different types may vary widely. However, we include the second variant so that a budget that covers all of an instance's children can be modeled in some fashion.

- **Tor Relay**: An instance is created for each relay in the consensus.
- **Relay Family**: An instance is created for each connected component of relays, where two relays are connected if they mutually reference each other in the family section of their descriptors [11].
- **(Relay Family, Tor Relay)**: An instance of this relationship is created for each relay belonging to a given family.
- **Relay Software Type**: This attribute is added to each relay based on the operating system reported in the relay's descriptor.

2. Standard techniques [22] are used to construct an AS-level Internet routing map. Data that can be used to create such a map includes the CAIDA internet topology [8], the CAIDA AS relationships [7], and RouteViews [31]. This map is then used to create the following instances:
   - **Virtual Link**: An instance is created representing the path between each Autonomous System and possible guard as well as between each Autonomous System and exit. A possible guard is a Tor relay that satisfies the requirements to serve as an entry guard. Guards and exits are determined from the Tor consensus. A virtual-link instance represents both directed paths between the Autonomous System and relay, which may differ due to Internet route asymmetries [15].
   - **AS**: An instance is created for each AS observed in the RouteViews data.
   - **(AS, Virtual Link)**: An instance of this relationship is created for each AS that appears on the path in either direction between the virtual link's AS and its relay, as determined by the Internet routing map.
3. Internet Exchange Points (IXPs) are added to paths in the AS-level Internet map based on data from the IXP Mapping Project [3]. These additions are used to create the following instances:
   - **IXP**: An instance is created for each IXP that appears on at least one path in the Internet map.
   - **(IXP, Virtual Link)**: An instance of this relationship is created for each IXP that appears on the path in either direction between the virtual link's AS and its relay, as determined by the Internet routing map.
4. ASes are clustered into organizations using the results of Cai et al. [5], and IXPs are clustered into organizations using the results of Johnson et al. [22]. Each cluster represents a single legal entity that controls multiple ASes or IXPs, such as a company. The clusters are used to create the following instances:
   - **AS Organization**: An instance is created for each AS cluster.
   - **IXP Organization**: An instance is created for each IXP cluster.
   - **(AS Organization, AS)**: An instance of this relationship is created for each AS in a given AS cluster.
   - **(IXP Organization, IXP)**: An instance of this relationship is created for each IXP in a given IXP cluster.
5. The system provides physical locations and legal jurisdictions for several of the ontology types. IP location information, such as from the MaxMind GeoIP database [26], provides location information for entities with IP addresses. The location of IXPs is frequently available on the Web as well [3]. The bilateral MLATs that might apply are obtained from the MLAT.is database [9]. These data are used to create the following instances and attributes:
   - **Legal jurisdiction**: An instance of this type is created for each country.
   - **(Legal jurisdiction, Relay)**: An instance of this relationship is created for each relay in a given country, as determined by the relay's IP address and the IP location information.
   - **(Legal jurisdiction, IXP)**: An instance of this relationship is created for each IXP in a given country, as determined by the IP addresses of the IXP or other public IXP information.
   - **Physical location**: This attribute is added to each relay with its geographic coordinates (i.e., latitude and longitude), as determined from its IP address. This attribute is also added to each IXP with its geographic coordinates, based on its IP addresses or other public IXP information.
   - **MLAT**: As a preliminary step, for each country instance C, create a duplicate country instance C', and add a relationship instance from C' to C. For each in-force, bilateral MLAT, create an MLAT instance (we assume there is at most one per pair of countries). For each MLAT instance M, if C_1 and C_2 are the instances of the two countries involved in the corresponding MLAT, add relationship instances from C_1' and C_2' to M and from M to C_1 and C_2. The duplicate C' instances will be the initially compromised ones. The structure described here will propagate this compromise to the original C instances, either directly or through MLATs. Here, we take the default effectiveness to be 1, i.e.,

each country always compromises its MLAT partners, but this may be changed by the user on a per-MLAT basis.

6. Although the system does not provide information about physical connections in general, it can use a cable database such as the TeleGeography database [29] or Greg's Cable Map [25] to add a cable instance for each cable in the database. It would still be left to the user to identify which virtual links use which cables, although incorporating this into the system is a topic of ongoing work.

4 Beliefs and BBNs

The user may provide various data to inform the operation of the system. However, many users may not wish to do this, and the system includes a default belief set designed to provide good security for average users. In Section 6 we describe a possible default belief set. For simplicity, we refer to beliefs as being provided by the user, but wherever they are not, the defaults are used instead.

4.1 User beliefs

Broadly, users may have two kinds of beliefs: those about the structure of the network, etc., and those about trust. The user's structural beliefs are used to edit the system-generated world to produce an "edited world;" we expect this will be done once, not on a per-adversary basis. These beliefs may describe new types and the addition or removal of type instances and relationships between them (e.g., adding relay operators known to the user). The user may also define new attributes, change the system-provided attributes, or provide values for empty attributes (e.g., labeling countries by their larger geographic region).

The user's beliefs may incorporate boolean predicates that are evaluated on instances in the revised world. For example, the user may have increased trust in ASes above a certain size. We sketch a suitable language for this in App. A, but this can be replaced with another if desired.

A user may have structural beliefs about instances of types and edges from the ontology. For types, a user may believe that an instance of that type exists; her belief about that instance must include a unique identifier for the instance and any required attributes. This type instance is then added to the system-generated world. The type of the instance may be system-generated, in which case this belief represents an edit to the system-generated world, or it may be user-generated. If the instance's type is user-generated, then the user must describe to the system how the instance should be translated to the BBN that the system produces from the edited world.

For edges, a user may believe that one type instance is the parent of another type instance. Her belief about such a relationship must include any required attributes of the corresponding edge type in the ontology. This relationship instance is then added to the system-generated world. If the edge type is not part of the ontology, the user must describe how the edge affects the computation of values in the BBN that the system produces.

Finally, the user provides trust beliefs of four types that are used in constructing the BBN from the revised world. The first two types of trust beliefs concern the propagation of compromise. Budget beliefs allow the user to say that an instance I in the edited world has the resources (monetary or otherwise) to compromise k of its children that satisfy some predicate P. Enforcing this as a hard bound appears to be computationally harder than we are willing to use in the BBN, so we do this in expectation. Compromise-effectiveness (CE) beliefs allow the user to express some correlations between the compromises of nodes by saying that, if an instance I is compromised, then, with probability p, all of I's children satisfying a predicate P are compromised. For example, this captures the possibility that a compromised AS compromises all of its routers except those of a particular model, for which the AS has made an error in their (common) configuration file.

The other two types of trust beliefs concern the likelihood of compromise. Relative beliefs allow the user to say that instances satisfying a given predicate (e.g., relays running a buggy OS, network links that traverse a submarine cable, or ASes that are small as determined by their number of routers) have a certain probability of compromise. (In particular, it specifies the probability that they remain uncompromised if they are otherwise uncompromised.) Absolute beliefs allow the user to say that instances satisfying a given predicate (e.g., the node is an AS and the AS number is 7007) are compromised with a certain probability, regardless of other factors.

4.2 Sample belief language

We now describe a sample language for users' structural and trust beliefs. This incorporates predicates, which might be expressed using the predicate language just outlined. In general, we assume that there is a set $\mathcal{V}$ of values that the user may use to express levels of trust. We illustrate this here by taking $\mathcal{V}$ to be $\{\mathsf{SC}, \mathsf{LC}, \mathsf{U}, \mathsf{LT}, \mathsf{ST}\}$; we think of these as "Surely Compromised," "Likely Compromised," "Unknown," "Likely Trustworthy," and "Surely Trustworthy." Our examples will not rely on $\mathcal{V}$ having exactly five elements, but we think this is one natural way that users might think about their trust in network elements.

4.2.1 Structural beliefs

Let $\mathcal{R}$ be the set of relationship instances in the system-created world. $\mathcal{R}'$ will be $\mathcal{R}$ augmented with all of the user-specified relationships.

Novel types A user may define new types via expressions of the form $(''\texttt{ut}'', tname, struct_{req}, struct_{opt})$, where $''\texttt{ut}''$ is a string literal, *tname* is a string (the name of the type) that must be distinct from all other *tname* values the user specifies and from all elements of $\mathcal{T}$, and where $struct_{req}$ and *structopt* are both descriptions of data structures (these may be empty data structures, which might be indicated by `NULL`). We write $\mathcal{T}'$ for the set containing the elements of $\mathcal{T}$ together with all of the *tname* values provided by the user.

Type instances An ordered list of tuples (T, D, n) $T \in \mathcal{T}'$, D is a data structure that is valid for T, and n is a unique identifier among these tuples.3 We write $\mathcal{I}'$ for the set formed by augmenting $\mathcal{I}$ with these new instances.

Relationship instances A set of pairs (P, C), where P (parent) and C (child) are type instances from $\mathcal{I}'$.4 We do not need to specify new relationship types, only the additional relationship instances.

4.2.2 Trust beliefs

Relative beliefs These are beliefs of the form (s, P, v), where s is a string other than $''\texttt{abs}''$, P is a predicate on factor variables, and $v \in \mathcal{V}$.

Note that, in our translation procedure below, relative beliefs affect the probability of compromise of a factor variable in the BBN that is not otherwise compromised through the causal relationships captured in the world.

Absolute beliefs These are beliefs of the form $(''\texttt{abs}'', \mathsf{P}, v)$, where P is a predicate on factor variables and $v \in \mathcal{V}$. A belief such as this says that the chance a variable satisfying P is compromised is captured by v. Note that it is the user's responsibility to ensure that no two different absolute beliefs have predicates that are simultaneously satisfied by a node if those beliefs have different values for v. We do not specify what value is used if this assumption is violated.5

Budget Expressed as either $(''\texttt{bu1}'', I, T, k)$ or $(''\texttt{bu2}'', I, \top, k)$, where $''\texttt{bu1}''$ and $''\texttt{bu1}''$ are string literals, I is a type instance in the edited world, T is a type in the edited world, and k is an integer. The interpretation is that, in expectation, compromise of the type instance with a Budget attribute will lead to compromise of k of its children (of type T in the first variant, or of all its children in the second variant).

Compromise effectiveness Expressed as either $(''\texttt{ce1}'', I,$ $\mathsf{P}_{\mathsf{ce}}, v)$ or $(''\texttt{ce2}'', I, \top, v)$, where $''\texttt{ce1}''$ and $''\texttt{ce2}''$ are string literals, I is an instance of a non-output type in the edited world, P_{ce} is a predicate on instances of a fixed type, $\top$ is a distinguished symbol, and $v \in \mathcal{V}$. The interpretation is that, if instance I is compromised, then it compromises its children satisfying P_{ce} (or all children, if $\top$ is given) with probability corresponding to v.

The actual probabilities that a compromised network element compromises other elements it controls, which CE beliefs attempt to capture, may tend to fall in a different range than other probabilities of compromise. Our translation procedure could be modified to treat the value v in a CE belief as a different probability than is used for other types of beliefs. Similarly, the belief language could be modified to

3 We assume that the system provides unique identifiers for the system-generated type instances and that the values of n in the user's list of tuples are distinct from those identifiers.

4 We abuse notation and use P and C in place of the unique identifiers associated with each type instance in the edited world.

5 A natural approach is to allow the use to specify these in an ordered list and using the last satisfied predicate.

allow CE beliefs to include probability instead of a value from $\mathcal{V}$.

4.2.3 Five-valued example

The following examples of beliefs illustrate how a user might express her beliefs in our five-valued example language.

1. Countries in set S_1 are likely trustworthy.
2. Countries in set S_2 are likely compromised.
3. Countries in set S_3 are surely compromised.
4. AMS-IX points are likely trustworthy.
5. MSK-IX points are of unknown trustworthiness.
6. Relay family F_1 is likely compromised.
7. Relay family F_2 is surely uncompromised.
8. Relay operator O_1 is surely uncompromised.
9. Relay operator O_2 is likely uncompromised.
10. Hosting company H_1 is surely trustworthy.
11. Submarine cables are of unknown trustworthiness.
12. Wireless connections are likely compromised.
13. Relays running Windows are of uknown trustworthiness (the system gets OS information from relay descriptors).
14. If an AS is compromised, then it is expected to be able to compromise 4 of the links that it is on.

We suggest that the compromise probabilities corresponding to the values SC, LC, U, LT, and ST might be taken by the sytem to be 0.999, 0.85, 0.5, 0.15, and 0.02, respectively. However, the user would express her beliefs in terms of "surely compromised," etc., as above. Whatever language is used to express beliefs, there would need to be an appropriate interface for users to express or import beliefs.

4.3 Translations to BBNs

A translation procedure in general needs to take the edited world (reflecting the structural beliefs and attribute values provided by the user) and the user's trust beliefs as input and produce a BBN as output. The output variables of the BBN should match the nodes in the edited world that are instances of types designated as output types in the ontology or the user's structural beliefs. Here, we present a translation procedure that fits with the rest of the system we describe (it matches our particular ontology, etc.).

As a component of our system, BBNs have both strengths and weaknesses. Their general strengths of being concise, being efficiently sampleable, and allowing computation of other properties of the distribution (e.g., marginal probabilities and maximum likelihood values) are beneficial in our system. BBNs are especially well-suited to our approach here because of the close structural similarity between our revised worlds and the BBNs we construct from these.

As a disadvantage, BBNs do not represent hard resource constraints efficiently; we can only approximate those here by constraining resources in expectation. More generally, other negative correlations may be difficult at best to capture, but it is possible that users will hold beliefs that imply negative correlations between compromise probabilities.

The purpose of this system is to produce an efficiently sampleable representation of compromise probabilities. Other representations of distributions could also be used, but they might be most naturally generated from trust beliefs in different ways. A detailed discussion of such approaches is beyond the scope of this work.

4.3.1 Our translation procedure

We now describe a translation procedure for the ontology and beliefs that we have presented above. Let W' be the final world that appears in the construction sequence described above.

- For each node (type instance) in W', the BBN contains a corresponding factor variable/node. We refer to the BBN node by the same name as the node in W'.
- For each compromise-effectiveness belief $B = (s, n, \mathsf{P}, v)$ about a node n, there is a corresponding child v_B of n in the BBN. The table for v_B is such that, if n is uncompromised, then v_B is uncompromised; if n is compromised, then v_B is compromised with probability $p(v)$ and uncompromised otherwise. (We use $p(v)$ to denote the probability value that the system assigns to the value $v \in \mathcal{V}$ that is part of the user's belief language.) The children of v_B in the BBN are the nodes in the BBN that correspond to nodes in W' that (1) are children of n and (2) satisfy the predicate P from the belief B. Assign these edges the weight set $\{1\}$; this auxiliary information will be used to construct the BBN's probability tables.
  If there are children of n in W' that do not satisfy any of the predicates in the compromise-effectiveness beliefs about n (including, e.g., when the user has no compromise-effectiveness beliefs), then make these nodes children of n in the BBN. Assign to each of

these edges the singleton weight set whose element is the appropriate default probability.6

– For each budget belief $B = (s, n, \mathsf{P}, k)$ about a node n, let $c_{n,\mathsf{P}}$ be the number of children of n (in W') that satisfy P. For each of these children, in the BBN, replace the single value in the edge's weight set by that value multiplied by $k/c_{n,\mathsf{P}}$.
– Assign to each non-CE-belief node n a "risk set" R_n that is initially empty. We add to R_n values that describe additional risk of n being compromised: For each belief $B = (s, \mathsf{P}, v)$ that has not already been evaluated and whose initial entry is not $''\texttt{abs}''$, if n satisfies P, then add v to R_n (retaining duplicates, so that R_n is a multiset).
– Construct the tables for each non-CE node in the BBN. (We have already constructed the tables for the CE-belief nodes.) Let n be a non-CE node. For each subset $\mathcal{S}$ of n's parents, if S is the multiset of weights on the edges from nodes in $\mathcal{S}$ to n, and if R is the multiset of risk weights associated with n, then the probability that n is compromised given that its set of compromised parents is exactly $\mathcal{S}$ is:

$$1 - \left(\prod_{p \in S}(1-p)\right)\left(\prod_{q \in R}(1-q)\right).$$

Note that, if the user has no parents, then the first product will be empty (taking a value of 1), and the probability of compromise will be determined solely by the risk factors unless the user expresses beliefs that override these.
– If the user provides a belief $B = (''\texttt{abs}'', \mathsf{P}, v)$, then nodes satisfying P are disconnected from their parents. Their compromise tables are then set so that they are compromised with probability $p(v)$ and uncompromised with probability $1 - p(v)$. This allows a user to express absolute beliefs about factor variables in the BBN (hence "abs"). In particular, she may express beliefs about input variables whose compromise would otherwise be determined by their attributes.

6 We assume that there are default values—perhaps just a single, common one—for the probability that the compromise of a node leads to the compromise of its children. These values might naturally depend on the types involved. Here, we suggest 1 as a common default value.

4.3.2 Potential extensions

We assume that adversaries are acting independently, although this may not always be the case. One natural example of inter-adversary dependence occurs with the compromise of resource-constrained instances in the world. For example, an ISP's resources may limit it to monitoring k of its routers. If both the ISP and the country (or other legal jurisdiction) controlling it are a user's adversaries, then they should compromise the same set of the ISP's routers. (This is true whether we model this compromise probabilistically, with k routers compromised in expectation, or through some other means.) This might be modeled statically by changing the structure of the BBN, but dynamic compromise and more general inter-adversary dependence may require other approaches.

At this point, our system does not include instances in the world in constructs that correspond to cities or states/provinces. These are most naturally viewed as instances of legal jurisdictions, and the user may well have beliefs about the corresponding laws or enforcement regimes. One way that we envision the user may address these is by adding to the world instances of legal jurisdictions that carry a "Boundary" attribute, effectively a predicate that can be evaluated on the system-provided geolocation data. The system could then determine which network entities are in which of these user-supplied jurisdictions. Physical locations might be handled this way as well, as long as the location is "large enough" relative to the resolution of the geolocation process.

5 Trust

We now discuss where trust judgments might originate. First, we present the rationale behind a trust policy that might be distributed with Tor client software as a default. Such a policy would be designed not to offer the best protection to particular classes of users but to adequately protect most Tor users regardless of where they are connecting to the network or what their destinations and behaviors are. Second, we discuss other sources of trust information and some use cases.

The most useful information about Tor relays for setting a default level of trust is probably relay longevity. Running a relay in order to observe traffic at some future time or for persistent observation of all traffic requires a significant investment of money and

possibly official authorization approval. This is all the more true if the relay contributes significant persistent capacity to the network. Further, operators of such relays are typically more experienced in many senses and thus somewhat less open to external compromise via hacking. The amount of relay trust is thus usefully tied to the length of presence in the network consensus, uptime, and bandwidth. This approach does not resist arbitrary large-budget, nation-state-scale adversaries with authority to monitor relays persistently, but it will help limit attacks to adversaries with such persistent capabilities and intentions. Resistance to particular nation-state adversaries would not make sense as a default trust policy for all Tor users worldwide.

There is no general reason to trust one AS, IXP, etc., more than another, but one should not presume that they are all completely safe. It thus is reasonable to assume the same moderate risk of compromise for all elements forming the links to the Tor network and between the relays of the network when creating a default trust policy. Though uniformly distributed, trust in these elements still plays a role in route selection. For example, a very high uniform level of trust would permit selection of routes through the same IXP if the trust in the selected relays themselves were found to be adequate. A lower level of trust might dictate a selection despite the availability of higher-longevity relays because of the AS or IXP risk. Note that moderating AS and IXP trust can also mitigate persistent nation-state adversaries to some extent if we assume individual ASes and IXPs are more likely to be compromised by the countries in which they are located.

Note that the average client using a default trust policy may be subject to errors because the average client's beliefs will rarely be exactly at the default. For any policy a client uses, the client may be subject to errors in the judgments that underly the policy.

Users with particular concerns might use non-default beliefs. These could be provided by, e.g., government entities, privacy organizations, political groups, media organizations, or organizations defending abuse victims. An example of an important non-default case is connecting users to sensitive destinations that they especially do not want linked to their location or possibly to their other Tor behaviors. For example, some users need to connect to sensitive employer hosts, and dissident bloggers could be physically at risk if seen posting to controversial sites. These users may have rich trust beliefs (either of their own or supplied by their organizations) about particular relays, ASes, etc., based on who runs the relay, hardware, location, etc.

6 Modeling a Network Adversary

We illustrate the use of our trust framework by considering a powerful, pervasive adversary called *The Man*. This adversary follows the suggestions in Sec. 5 and thus is a plausible candidate for a default trust belief in Tor. We construct The Man by drawing on a variety of public data sets and evaluate its ability to compromise users' paths.

6.1 Constructing The Man

We allow The Man to compromise relay families and AS or IXP organizations, where a family or organization is a group controlled by the same entity. Each family is compromised by The Man independently with probability between 0.001 and 0.1, where the probability increases as the family's longevity in Tor decreases. Specifically, the probability of compromise for a family f with uptime u_f was set to be $(0.1 - (0.1 - 0.001))u_f$. Each AS and IXP organization is compromised independently with probability 0.1.

To construct The Man adversary, we must create a routing map of the Internet that includes ASes, IXPs, and Tor relays. We must also group ASes and IXPs into organizations, identify relay families, and evaluate the longevity of Tor relays. We do so using the techniques and data sources described in Section 3.3.

To build the routing map, we use CAIDA topology and link data from 12/13 and RouteViews data from 12/1/13. The resulting map includes 46,368 ASes, 279,841 links between ASes, and 240,442 relationship labels. To group ASes by the organization that controls them, we use the results of Cai et al. [5]. These contain data about 33,824 of the ASes in our map, and they result in 3,064 organizations that include more than one AS with a maximum size of 81 and a median size of 2. We use the results of Augustin et al. [3] to identify IXPs and their locations between pairs of ASes. These results show 359 IXPs and 43,337 AS-pairs between which at least one IXP exists. We then use the results of Johnson et al. [22] to group IXPs into organizations. These produce 19 IXP organizations with more than one IXP, for which the maximum size is 26 and the median size is 2.

We add relays to the routing map using Tor consensuses and descriptors from Tor Metrics [30]. We use the Tor consensus of 12/1/13 at 00:00. The network at this time included 1,235 relays that were guards only, 670 re-

lays that were exits only, and 493 relays that were both guards and exits. The consensus groups relays into 152 families of size greater than one, of which the maximum size was 25 and the median size was 2. Family uptime is computed as the number of assignments of the Running flag to family members, averaged over the family members and the consensuses of 12/2013. We map the Tor guards and exits to ASes using Routeviews prefix tables from 12/1/13, 12/2/13, and 11/30/13, applied in that order, which is sufficient to obtain an AS number for all guards and exits. Note that we observe one exit relay that mapped to an AS that does not appear in our map, and so we add that additional AS. There are 699 unique ASes among the guards and exits.

We create paths from each AS in our map to each guard and exit AS. The median number of paths that we can infer to a guard or exit AS is 46,052 (out of the 46,369 possible). The maximum AS path length is 12, and the median AS path length is 4. The maximum number of IXPs on a path is 18, and the median number is 0.

The resulting BBN for The Man thus includes 2398 relay variables (one for each guard and exit) and 32,411,931 virtual links (one from each AS to each guard or exit AS). For any path missing from our routing map, we simply take the path to include only the source AS and destination AS.

6.2 Analysis

We consider security from 58 of the 60 most common client ASes as measured by Juen [24] (AS8404 and AS20542 do not appear in our map). Juen reports that these 58 ASes covered 0.951 of client packets observed.

For each of our 58 client locations, we choose an exit and guard using Tor's path-selection algorithm as implemented in TorPS [22]. Note that (among other considerations) this does ensure that the guard and exit don't share the same family or /16 subnet. Then we sample The Man BBN to determine if the resulting circuit to the server is vulnerable to a first–last correlation attack.

Over 100,000 trials, the minimum, mean, median, and maximum probabilities of compromise were 0.108, 0.132, 0.127, and 0.164, respectively.

7 MLATs and Their Effects

Our ontology described above includes Mutual Legal Assistance Treaties (MLATs), which have not received significant previous attention in the study of attacks on Tor. If a user of our trust system is worried about state-level adversaries, then MLATs are potentially significant. To illustrate this, we do some preliminary analysis of the effects of MLATs on the ability of (randomly constructed, composite) state-level adversaries to carry out first–last correlation attacks. We start with an overview of MLATs and the data we use about them.

7.1 MLATs

MLATs formally require and enable their signatories to cooperate in many aspects of criminal legal assistance, from investigations, to collection of evidence, to extradition of targets or suspects. Most relevantly to the consideration of adversary power, this enables countries, through MLATs, to gain information from network components in other legal and governmental jurisdictions.

MLATs have existed since ancient times. In the last half century, the number of MLAT relationships between countries has grown sharply, and the number of countries that participate in MLATs at all has also increased [10].

Currently, thousands of MLATs are in some stage of negotiation or entry into force between different countries. In order to analyze the effects of MLATs on first–last compromise in Tor, we make use of the MLAT database behind the `www.MLAT.is` [9] site described by Cortes [10]. This draws on treaty data from a variety of original sources, such as national governments and international organizations. It analyzes when and whether the treaties are actually in force, the countries subject to them, and what type of applications (e.g., extradition) the treaty has.

MLATs vary in their strength, such as what kinds of exceptions they include, the strength of evidence collection they cover, and the extent to which one partner can coerce another to share information. While an increasing body of international case law exists with respect to MLATs, some MLATs remain untested in terms of how easily they can be used to get from a treaty partner information that is potentially covered by the MLAT. A country may be able to influence one of its MLAT partners—i.e., a country with whom it has a direct treaty—in invoke one of the partner's treaties with

yet other countries in order for the original country to obtain more information. Such scenarios have less well-defined parameters and offer the original country less direct power. In our analysis in Sec. 7.4, we thus focus only on direct relationships instead of the transitive application of MLATs. We also restrict our attention to the MLATs that are most likely to be effective in the setting we consider here, namely the non-extradition, bilateral, criminal MLATs that are already in force. However, our framework can easily be adapted to consider, e.g., transitive applications of MLATs or multilateral MLATs.

7.2 Cable data

In order to study the effects of MLATs on the power of adversaries, and to account for submarine cables as a network-component type of new interest, we randomly construct composite state-level adversaries ("pseudo-countries") that comprise countries appearing in cable and MLAT data. As described in Section 7.3, this incorporates weighting by cable bandwidth and accounting. As noted above, we study the ways that in-force, bilateral, non-extradition, criminal MLATs expand the capabilities of the pseudocountries that we construct. The version of the MLAT.is data that we use provided 559 applicable MLATs.

We use Greg's Cable Map (`cablemap.info`) [25] as our source for cable data. This includes bandwidth information for many of the cables. Other sources, such as TeleGeography's data [29], could be used instead or as well once bandwidth information was added.

After some cleanup of the data as described in App. B, we were left with 222 cables with a total bandwidth of over 722,000 Gb/s. Different data sources may differ in the exact set of cable systems that they cover, and bandwidth data may be reported differently by different sources or be unavailable. However, we believe the data we use are plausible, and they certainly demonstrate the feasibility of the approach we present here.

To determine a country's cable bandwidth, we counted the total bandwidth of all cables with landing points coded in that country. We use "MLAT reach" to mean the total bandwidth of all cables landing in a country or its MLAT partners. Figure 2 shows the countries with at least 550,000 Gb/s in MLAT reach. The top bar shows the total bandwidth of *all* cables in the data set; the country-specific bars show each country's bandwidth (light part of the bars) and MLAT reach (light and dark parts of the bars together). The ranking of countries by MLAT reach is notably different than their ranking by bandwidth (see App. B).

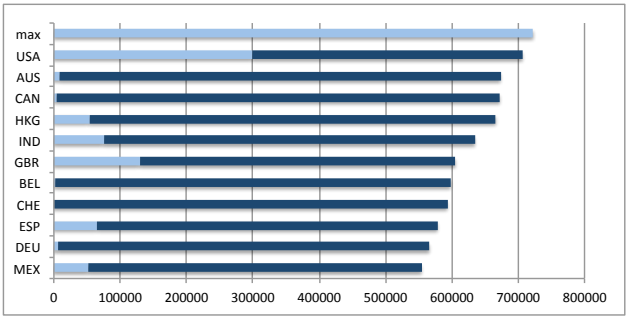

Fig. 2. The amount of cable bandwidth (Gb/s) controlled by countries directly (light part of bars) and in collaboration with their first-degree MLAT partners (entire bars) for the 11 countries that control at least 550 Tb/s in collaboration with their MLAT partners. The top bar shows the total bandwidth of all cables in the data set.

7.3 Pseudocountry construction

We use the submarine-cable data to construct random, hyptothetical adversaries ("pseudocountries") comprising countries that appear in the cable and MLAT data sets. While we believe the data we have obtained are generally plausible, using hypothetical adversaries allows us to analyze adversary capabilities without being distracted by issues surrounding the precise capabilities of real countries.

In order to construct our pseudocountries, we use a randomized procedure that iteratively picks new countries, weighted by country bandwidth, and adds them to the pseudocountry if they satisfy constraints that parameterize the procedure. Figure 3 in App. C shows pseudocode for this procedure.

As a preliminary step to randomly constructing the pseudocountry adversaries that we use here, we considered various combinations of constraints. For each candidate constraint combination, we ran numerous random trials and evaluated the resulting pseudocountries in terms of the amount of the bandwidth and number of cables that they control directly and might be able to access through their first-degree MLAT partners. This suggested the constraint combinations that we use here to get pseudocountries that control large, medium, and small amounts of cable bandwidth, both directly and with their MLAT partners. We treat a country C as an MLAT partner of a pseudocountry P if any one of the constituent countries in P is listed in the MLAT

data as a partner of C. The constraints also vary in the number of MLAT partners that each resulting pseudocountry has. Table 4 in App. C provides statistical information about the pseudocountries generated by these constraints over 10,000 trials. After finalizing the constraint sets, we then ran the randomized pseudocountry-construction procedure *once* with each of these chosen combinations and took the resulting output as the adversaries that we use here.

The first constraint set we use is:

- Allow at most one country that directly sees at least 100,000 Gb/s (i.e., USA, COL, BRA, JPN, PRI, GBR, PAN, and CHN).
- Allow at most two countries that directly see at least 60,000 Gb/s but less than 100,000 Gb/s (i.e., ZAF, ECU, VGB, ABW, CYP, RUS, IND, SGP, ESP, and AGO).
- Allow at most five countries in total

The second constraint set we use is:

- Allow the total capacity seen directly by the pseudocountry to be at most 50,000 Gb/s.
- Allow the total capacity seen by the pseudocountry and its first-degree MLAT partners to be at most 120,000 Gb/s.

The third constraint set we use is:

- Allow no country that, together with its first-degree MLAT partners, sees at least 500,000 Gb/s (i.e., USA, AUS, CAN, HKG, IND, GBR, BEL, CHE, ESP, DEU, MEX, ITA, UKR, ARG, SVK, BRA, ROU, SVN, ZAF, POL, HUN, GRC, and FRA).
- Allow the total capacity seen directly by the pseudocountry to be at most 400,000 Gb/s.
- If a new country (after the first one) is not an MLAT partner of one of the countries already in the pseudocountry, then either the new country or all of the existing countries must have no MLAT partners at all.
- Allow at most four countries in total.

After deciding on these constraint sets using the statistics about the countries they randomly generated, we then randomly generated one pseudocountry (P_1, P_2, and P_3, respectively) from each constraint set for use as an adversary. Those pseudocountries and the pseudocountries together with their MLAT partners (denoted M_1, M_2, and M_3) are:

- P_1: AUS, ESP, GTM, JPN, and SGP
- M_1: ARE, ARG, AUS, AUT, BEL, BOL, BRA, CAN, CHE, CHL, CHN, COL, CPV, CZE, DOM, DZA, ECU, ESP, FIN, FRA, GBR, GRC, GTM, HKG, HUN, IDN, IND, ISR, ITA, JPN, KAZ, KOR, LUX, MAR, MCO, MEX, MRT, MYS, NLD, PAN, PER, PHL, PRT, PRY, SGP, SLV, SVN, SWE, THA, TUN, URY, USA, and ZAF
- P_2: ALB, ASM, DJI, GUF, GUM, HND, MNE, MSR, MUS, PNG, SLE, SYR, TZA, and WSM
- M_2: ALB, ASM, BEL, CZE, DJI, GRC, GUF, GUM, HND, HUN, IND, MNE, MSR, MUS, PNG, POL, ROU, SLE, SVK, SYR, TZA, and WSM
- P_3: ECU, FJI, GUM, and QAT
- M_3: ARG, AUS, BOL, CHE, ECU, FJI, GBR, GUM, and QAT

	P_1	M_1	P_2	M_2	P_3	M_3
# Countries	**5**	**53**	**14**	**22**	**4**	**9**
Capacity (10^3 Gb/s)	**287**	**678**	**50**	**112**	**112**	**250**
Capacity (%)	**39.8**	**93.8**	**6.9**	**15.4**	**15.4**	**34.6**
Cables (out of 222)	**56**	**189**	**32**	**53**	**19**	**68**

Table 1. Pseudocountry characteristics.

Table 1 shows statistics about the P_is and M_is including the number of constituent (real) countries, the cable capacity that they control (both absolute bandwidth and the fraction of the total cable bandwidth), and the number of cables they control out of the 222 in our data set.

7.4 Analysis

We study the effects of MLATs on compromise by considering a variety of ways in which an adversary can cooperate with its MLAT partners. The differences reflect both various possible levels of coordination between a country and its MLAT partners as well as the potential difficulties with actually enforcing MLATs. The compromise models we consider are the following:

Type P: The adversary compromises a path exactly when the pseudocountry appears on both virtual links in a path (i.e., between the source and the guard and between the exit and the destination).

Type P+M: The adversary compromises a path exactly when the pseudocountry P appears on both virtual links or there is an MLAT partner M of P that appears on both virtual links. This models M sharing the results of its unilateral attacks with P.

Type P+PM: The adversary compromises a path exactly when the pseudocountry P appears on both virtual links or there is an MLAT partner M of P such that P appears on one virtual link and M appears on the other. This models M sharing information with P that produces a coordinated attack but not sharing the results of its unilateral attacks.

Type P+M+PM: The adversary compromises a path exactly when the pseudocountry P appears on both virtual links or there is an MLAT partner M of P such that either M appears on both virtual links or P appears on one virtual link and M appears on the other. This models M both sharing partial information to produce a joint attack and information about M's unilateral attacks.

For all types of coordination, we can also consider the effects using each MLAT partner with a specified probability p. Individual MLATs may turn out to be difficult to use for a particular application or take too long to apply. We model this by fixing a probability $p = 0.5$ and, for each MLAT partner country C, including C in the adversary with probability p. (In particular, if C is included in the adversary, it is included for all paths.) This probability could easily be varied, or a different probability of enforcement could be assigned to each MLAT. For example, this would allow users who have beliefs about the degree to which different MLATs are effective (e.g., accounting for the factors discussed at the end of Sec. 7.1) to incorporate those beliefs into this framework.

We use data obtained by Juen et al. [23] containing traceroutes from a selection of Tor relays to destinations randomly chosen from approximately 500,000 address blocks. We did not discard incomplete traceroutes, but we did omit traceroutes whose source relays were listed in 192.168.*.* and 10.*.*.*. This left us with paths from 57 relays to destinations across the Internet. We used the GeoIP database [26] to geolocate the addresses in the traceroute data.

We consider whether a pseudocountry adversary, under different coordination models described above, could observe both virtual links (outside of the Tor network) of a path constructed by combining two paths from the traceroute data set that use different relays. This treats all of the Tor relays as both guards and exits, an assumption made necessary by the limited number of relays, and assumes symmetric routing. Our focus here is on the extent to which MLATs increase the reach of adversaries, so we consider absolute numbers of paths rather than weighting by relay bandwidths. Because of the volume of the data, we do not attempt to weed out paths that start and end at the same IP but use different relays. However, the *possible* effect of these paths on our results is at most one part in 10^5.

Table 2 shows, for the coordination models of interest noted above, the fraction of paths observed under each model by each of the three pseudocountries constructed in Section 7.3. It also shows the 10^{th}, 50^{th}, and 90^{th} percentiles of compromise fractions (over 10,000 trials) when each MLAT partner cooperates with probability 0.5. In all cases, MLATs can allow for much greater reach.

Pseudocountry→	1	2	3
Type P	0.019	0.000	0.000
Type P+M	0.66	0.005	0.098
Type P+PM	0.248	0.001	0.006
Type P+M+PM	0.737	0.006	0.103
Type P+M ($p = 0.5$; 10%)	0.042	0.000	0.000
Type P+M ($p = 0.5$; 50%)	0.596	0.003	0.002
Type P+M ($p = 0.5$; 90%)	0.650	0.005	0.098
Type P+PM ($p = 0.5$; 10%)	0.104	0.000	0.000
Type P+PM ($p = 0.5$; 50%)	0.192	0.001	0.001
Type P+PM ($p = 0.5$; 90%)	0.239	0.001	0.006
Type P+M+PM ($p = 0.5$; 10%)	0.127	0.0	0.001
Type P+M+PM ($p = 0.5$; 50%)	0.317	0.003	0.1
Type P+M+PM ($p = 0.5$; 90%)	0.72	0.006	0.102

Table 2. The fraction of paths in the universe considered that are compromised by each of the three pseudocountry adversaries with different types of coordination. Percentiles for probabilistic enforcement of MLATs are computed over 10,000 trials.

8 Using Trust to Improve Path Selection

As a proof of concept, we examine how trust might be used to improve security in Tor. In particular, we consider how trust might be used to prevent the first–last correlation attacks by The Man (introduced in Sec. 6) when accessing a given online chat service. We suppose that users use trust to choose paths that are less likely to be vulnerable to this attack and run experiments to evaluate how effective this might be. These experiments just show the potential for improvement from using trust; they do not take into account other attacks or how to maintain good performance.

8.1 Experiments

Against The Man, we examine both how users can choose more-secure paths through Tor and how the service can choose server locations to make them more securely accessible via Tor.

For our experiments, we use as the destination service the Web chat server `webirc.oftc.net`. This IRC service is run by the Open and Free Technology Commu-

nity and is popular with Tor developers. As in Sec. 6.2, we consider users coming from 58 of the top 60 client ASes measured by Juen. In addition, for all of our experiments, the compromise probability (i.e., the probability of a first–last correlation attack by The Man) is estimated by sampling from The Man BBN 100,000 times and using the fraction of compromised samples as the probability.

The algorithms we use in our experiments are as follows:

- **Clients use trust**: Guards are chosen for each client location to be the three relays with the smallest probabilities that the adversary compromises the guard or an AS or IXP on the path to the guard. Then for a given destination, the algorithm considers using each of the client location's three guards with each Tor exit relay, estimates the probability of first–last compromise, and chooses the guard and exit with lowest resulting probability.
- **Service uses trust**: We only consider each AS containing an exit relay as a possible location for the server because these locations have the minimal chance for the adversary to observe traffic between the exit and destination. For each potential server location, we compute the probability of first–last compromise for each client location. This is estimated for a given client location by considering each of its guards, considering each exit sharing the server location, estimating the compromise probability, and using the minimum of these probabilities. We choose the server location with the minimum average compromise over all client locations. We add each additional server greedily by repeating the same process except that we only update the compromise probability for a client location if it decreases when using the new potential server location.

8.2 Analysis

Our results are shown in Table 3. For ease of comparison, the first row shows the results presented in Sec. 6 for a client using Tor's default path-selection algorithm. We can see that by using trust to choose guard and exit relays, clients can reduce the compromise probability by a factor of over 2.8 on average. When in addition the service changes the location of its server, that probability drops again by a factor of over 2.7 and approaches the minimum possible of $(0.1)^2 = 0.01$. It appears that adding additional server locations does not add significantly to user security. Note that each probability is estimated with 100,000 samples, which can explain why some probabilities are slightly below 0.01 and why the probabilities sometimes increase slightly when a server is added.

	Mean	Median	Min	Max
Default Tor	**0.132**	**0.127**	**0.108**	**0.164**
Only client uses trust	**0.046**	**0.049**	**0.026**	**0.091**
Client+service, 1 server	**0.017**	**0.018**	**0.009**	**0.033**
Client+service, 2 servers	**0.017**	**0.017**	**0.009**	**0.034**
Client+service, 3 servers	**0.017**	**0.017**	**0.009**	**0.033**

Table 3. First–last correlation probabilities against The Man for 58 client locations

8.3 Discussion

Note that our evaluation of these algorithms serves as a proof of concept for how our trust framework might be used. We do not propose that the trust-based algorithms we evaluate should be used in Tor exactly as described. Choosing paths in Tor based on the underlying network topology potentially creates security vulnerabilities outside of first–last correlation. For example, as has been observed in other work using trust and network location [1, 20, 21], the adversary could place his own Tor relays in locations that a user is more likely to select, and identities of the relays on a path chosen based on a client's trust and location may themselves reveal information about the client. In addition, path selection must take load balancing into account in order for Tor to maintain adequate performance. We leave as an open problem designing path-selection algorithms that use trust and location information in a way that convincingly improves security while maintaining performance.

We also note that the server-location algorithm, while it may appear overly optimistic, does seem plausible in many important use cases. Services with few Tor customers or without enough resources to run multiple servers for a diverse client base do seem unlikely to be able to move their servers to improve security as much as we could in our experiments. However, we can imagine that users might run servers for personal use and choose to locate them in a way that keeps them secure against their adversaries. It also seems plausible that large organizations with privacy-conscious constituencies, such as banks or governments, could run multiple redundant servers in order for their clients to choose a location that can be accessed securely.

9 Conclusions and Future Work

In this paper we have outlined a general but practical approach to represent network trust for the purposes of anonymous communication. Our approach represents network trust as an arbitrary probability distribution over the possible sets of network elements observed by the adversary, and we describe how to conveniently and efficiently represent and express such distributions. Our model allows trust in general network elements, not just relays, and our adversary distributions are more general than previously considered. This model can be used to inform the user's path selection in Tor, helping her to avoid first–last correlation attacks.

We have introduced two novel adversaries that can be expressed and analyzed using our trust framework. First, we presented The Man, which represents a pervasive, powerful adversary that we have argued could be used as the default trust belief in Tor. Second, we discussed the risk of Mutual Legal Assistance Treaties (MLATs), both incorporating these into our system and analyzing their effects on adversary capabilities under different models of coordination between the adversary and its MLAT partners.

We have also carried out preliminary experiments that show that the potential for our notion of trust to reduce the probability of first–last correlation against a client who uses trust to inform path selection. This probability of attack is further reduced when the service accessed by the user has positioned its server(s) in a way that is also informed by our work.

Ongoing and future work includes the further development and investigation of Tor path-selection algorithms that use trust as formalized here, the further development and analysis of methods to express trust that are natural and usable, the continued analysis of possible trust errors and their effects, and the development of a user interface for importing or entering trust beliefs. Two particularly important tasks are the development of collections of trust beliefs that capture important use cases and the study of how users can use different trust beliefs without being identified by that behavior.

Another future direction for research is making the MLAT model more complex, accounting for multilateral MLATs and variations in enforcement probability between different MLATs. Our focus in considering MLATs was their effect on the adversary's reach as a fraction of paths, but this question would also be interesting in a non-uniform setting corresponding to Tor's usage patterns. Another topic of ongoing and future work is the modeling of the adversary's control of individual submarine cables.

Acknowledgments

This expands upon a preliminary version that was presented without formal proceedings [19]. This work was supported in part by NSF grant 1016875 and in part by the Defense Advanced Research Agency (DARPA) and SPAWAR Systems Center Pacific, Contract No. N66001-11-C-4018. The work at NRL was also supported by ONR.

References

[1] M. Akhoondi, C. Yu, and H. V. Madhyastha, In: S. Jha and W. Lee (Eds.), 2012 IEEE Symposium on Security and Privacy, May 21–23, 2012, San Francisco, USA (IEEE Computer Society, Los Alamitos, 2012) 476–490, DOI:10.1109/SP.2012.35

[2] Alcatel-Lucent, http://www.alcatel-lucent.com/press/2013/002779, accessed October 22, 2014

[3] B. Augustin, B. Krishnamurthy, and W. Willinger, In: A. Feldmann and L. Mathy (Eds.), 9th ACM SIGCOMM Internet Measurement Conference, November 4–6, 2009, Chicago, USA (ACM, New York, 2009) 336–349, DOI:10.1145/1644893.1644934

[4] A. Biryukov, I. Pustogarov, and R.-P. Weinmann, In: W. Lee, A. Perrig, and M. Backes (Eds.), 2013 IEEE Symposium on Security and Privacy, May 19–22, 2013, San Francisco, USA (IEEE Computer Society, Los Alamitos, 2013) 80–94, DOI:10.1109/SP.2013.15

[5] X. Cai, J. Heidemann, B. Krishnamurthy, and W. Willinger, USC/Information Sciences Institute Technical Report ISI-TR-2009-679, http://www.isi.edu/~johnh/PAPERS/Cai12b/index.html

[6] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson, In: G. Danezis and V. Gligor (Eds.), 2012 ACM Conference on Computer and Communications Security, October 16–18, 2012, Raleigh, USA (ACM, New York, 2012) 605–616, DOI:10.1145/2382196.2382260

[7] CAIDA, http://www.caida.org/data/as-relationships/

[8] CAIDA, http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml

[9] S. Cortes, Database supporting http://www.mlat.is, accessed November 3, 2014.

[10] S. Cortes, Rich. J.L. & Tech. 22 (2015) (in press) SSRN abstract available at http://ssrn.com/abstract=2564218.

[11] R. Dingledine and N. Mathewson, https://gitweb.torproject.org/torspec.git/blob_plain/HEAD:/path-spec.txt, accessed February 2014

[12] T. Elahi and I. Goldberg, University of Waterloo CACR Technical Report CACR 2012-33, http://cacr.uwaterloo.ca/techreports/2012/cacr2012-33.pdf

[13] ESB Telecoms, http://www.esbtelecoms.ie/emerald_bridge/overview.htm, accessed October 22, 2014
[14] J. Y. Halpern, Reasoning About Uncertainty (MIT Press, Cambridge, 2003)
[15] Y. He, M. Faloutsos, S. V. Krishnamurthy, and B. Huffaker, In: S. E. Watikins (Ed.), IEEE Global Telecommunications Conference, November 28–December 2, 2005, St. Louis, USA (IEEE, Piscataway, 2005) 904–909, DOI:10.1109/GLOCOM.2005.1577769
[16] Interchange, http://interchange.vu/benefits-for-vanuatu/, accessed October 22, 2014
[17] Interchange, http://interchange.vu, accessed October 22, 2014
[18] ISO 3166, Country codes
[19] A. D. Jaggard, A. Johnson, P. Syverson, and J. Feigenbaum, arXiv:1406.3583v1 [cs.CR], presented at HotPETs 2014
[20] A. Johnson and P. Syverson, In: J. Mitchell (Ed.), 22nd IEEE Computer Security Foundations Symposium, July 8–10, 2009, Port Jefferson, USA (IEEE Computer Society, Los Alamitos, 2009) 3–12, DOI:10.1109/CSF.2009.27
[21] A. Johnson, P. Syverson, R. Dingledine, and N. Mathewson, In: G. Danezis and V. Shmatikov (Eds.), 18th ACM Conference on Computer and Communications Security, October 17–21, 2011, Chicago, USA (ACM, New York, 2011) 175–186, DOI:10.1145/2046707.2046729
[22] A. Johnson, C. Wacek, R. Jansen, M. Sherr, and P. Syverson, In: V. Gligor and M. Yung (Eds.), 2013 ACM Conference on Computer and Communications Security, November 4–8, 2012, Berlin, Germany (ACM, New York, 2013) 337–348, DOI:10.1145/2508859.2516651
[23] J. Juen, A. Das, A. Johnson, N. Borisov, and M. Caesar, arXiv:1410.1823v2 [cs.CR]
[24] J. P. J. Juen, M.S. thesis, University of Illinois at Urbana-Champaign (Urbana-Champaign, USA, 2012)
[25] G. Mahlknecht, http://cablemap.info, accessed October 8, 2014
[26] MaxMind, http://dev.maxmind.com/geoip/legacy/geolite/
[27] Submarine Telecoms Forum, Inc., http://subtelforum.com/Issue11/, accessed October 17, 2014
[28] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr, In: H. Federrath (Ed.), Designing Privacy Enhancing Technologies (Springer Verlag, Heidelberg, 2001) 96–114, DOI:10.1007/3-540-44702-4_6
[29] TeleGeography, https://github.com/telegeography/www.submarinecablemap.com/
[30] The Tor Project, Inc., https://metrics.torproject.org/, accessed April 2014
[31] University of Oregon, http://www.routeviews.org/
[32] P. Winter and S. Lindskog, Spoiled onions: Exposing malicious Tor exit relays, arXiv:1401.4917v1 [cs.CR]

A A language for predicates

We expect that the user may want to express some of her beliefs (trust and perhaps also structural) in terms of predicates, even though she might not be able to effectively evaluate these herself. For example, the user's trust in ASes with very few routers might be different than her trust in ASes with many routers (perhaps because she believes that larger ASes are more likely to have processes, policies, and organizational experience that prevent misconfiguration). She might capture this with a predicate that expresses whether the number of routers in an AS (in the edited world) is at least as great as a specified threshold.

The belief languages must thus incorporate a language for predicates that the system can interpret. We treat the predicate language as a separate component, and we sketch here one predicate language that will be used by all of our example belief languages. This language includes:

Connectives and operators Basic logical connectives (including negation)

Typing Testing whether an instance or attribute is or is not of a specified type; users may test for types not in the ontology (e.g., to check types that they have added)

Sets Sets (explicitly enumerated or defined by some predicate) and set membership/non-membership

Membership A predicate may depend on a set and test whether a value is in that set.

Tests of attribute values Tests must be appropriate to the data type used in the attribute; equality and inequality tests are allowed unless specified otherwise. Predicates may test user-defined attributes.[7] This may reference user-defined attributes.

Tests of the world structure (trust beliefs only) After the world is constructed and edited (i.e., when applying trust beliefs but not when applying structural beliefs), we allow predicates in beliefs to refer to the structure of the world.

B Cable Data

As noted above, we did some minor cleanup of the cable data. As this work is intended to demonstrate the feasibility of our approach, we do not attempt to construct a definitive description of international submarine cables. The changes that we make to the original

7 We expect that user-defined attributes will only be tested by the user, e.g., through predicates that she specifies on those attributes. As noted in the construction sequence in Section 2.1, the system will not change the structure of the world based on user-defined attributes.

data demonstrate the flexibility of this system as more and newer information becomes available to a user. We note that different sources sometimes provide different descriptions of cables, especially with respect to bandwidth. As discussed below, we have attempted to provide capacity data for cables lacking it in the original data set.

We coded the landing points in the data using the ISO-3166 country list [18]; we followed the MLAT data set when questions arose of which should be considered independent. We omitted cables that the data appear to indicate are not live, go over land, or are no longer used for general Internet traffic. The coding of cable landings required a little cleanup to match the appropriate countries from the MLAT.is database. After this initial cleanup, we checked all cables that were listed as landing in only one country or that lacked a listed capacity.

For cables without capacities listed, the then-current Submarine Cable Almanac [27] provided values that we use as follows: BBG (Bay of Bengal Gateway), 30 Tb/s; TGN-Gulf, 1.28 Tb/s; ADRIA-1, 622 Mb/s; Suriname-Guyana (SG-SCS), 1.28 Tb/s; CeltixConnect, 960 Gb/s. For the America Movil-1 cable, we use a press release [2] as the source for a 50 Tb/s capacity. For the Emerald Bridge cable, the cable website notes that it has 96 fibers [13]; we take 9.6 Tb/s as a guess for its total capacity. For the Vanuatu-Fiji Interchange Cable Network, we use the 1.28 Tb/s figure from the cable website [16]. For cables Melita-1 and WARF, no capacity data appears to be available. We use non-zero guesses of 100 Gb/s for each of these cables (whose in-service dates from the original data set are 2009 and 2007, respectively).

For cables with fewer than two countries listed in the set of landing countries that we generate, we examine the cables in more detail. We generally omit those that are listed as landing in only one country. Exceptions Both FLAG Atlantic (FA-1) and FLAG ATLANTIC NORTH are listed as landing in a single country (USA and GBR, respectively); based on the Submarine Cable Almanac, we treat these as a single cable with landings in USA, GBR, and FRA. Based on the cable's website [17], we take the landings of the Vanuatu-Fiji Interchange Cable Network to be VUT and FJI.

Figure 4 shows the amount of total bandwidth of cables landing in countries that have at least 60,000 Gb/s of such bandwidth. We note that adding bandwidth from cables landing in first-degree MLAT partners significantly changes the bandwidth rankings. For example, COL, BRA, JPN, and PRI rank in positions two through five when considering only country bandwidth, but they do not rank in the top 15 in MLAT reach (compare with Fig. 2 in Section 7.2).

C MLAT Adversary Construction

Before generating the pseudocountries used in our analysis, we repeatedly tested possible constraint sets using the pseudocountry-generation algorithm described in Fig. 3. Table 4 shows statistics for each of the three constraint sets when used in 10,000 random trials. For each constraint, Tab. 4 shows the 10^{th}, 25^{th}, 50^{th} (median), 75^{th}, and 90^{th} percentiles, as well as the mean, over these random trials for the number of countries in the pseudocountry, the cable capacity seen directly by the pseudocountry, the number of countries in the pseudocountry together with its MLAT partners, and the MLAT reach of the pseudocountry. We used this type of statistical output to tweak our constraint sets. Once those sets were finalized, we then randomly generated one pseudocountry for each of the three constraint sets to obtain the adversaries described in Sec. 7.

repeat
  Select cable at random (bandwidth weighted)
  if No landing country on the cable is part of the pseudocountry **then**
    repeat
      Pick a landing country from the cable at random (bandwidth weighted)
      if Allowed by country constraints **then**
        Add that country to the pseudocountry
      end if
    until A country is added to the pseudocountry or no more countries left to try on this cable
  end if
until The maximum number of countries (15 if not otherwise specified in the constraints) has been added or there are no more cables to try. Restart the process if there are fewer than two countries.

Fig. 3. Pseudocode for constructing pseudocountries, parameterized by country constraints

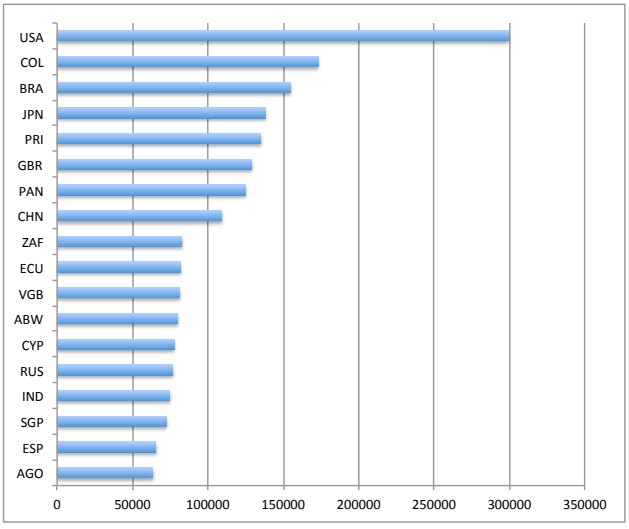

Fig. 4. The amount of cable bandwidth (Gb/s) controlled by countries directly for the 18 countries that control at least 60 Tb/s directly.

		Set 1	Set 2	Set 3
Countries	**10%**	**5.0**	**8.0**	**4.0**
	25%	**5.0**	**9.0**	**4.0**
	50%	**5.0**	**11.0**	**4.0**
	Mean	**5.0**	**11.1**	**4.0**
	75%	**5.0**	**13.0**	**4.0**
	90%	**5.0**	**15.0**	**4.0**
Capacities	**10%**	**247,498**	**49,843**	**164,429**
	25%	**311,760**	**49,924**	**205,263**
	50%	**357,764**	**49,984**	**251,436**
	Mean	**370,197**	**49,813**	**252,436**
	75%	**440,099**	**49,999**	**301,889**
	90%	**478,859**	**50,000**	**340,469**
Partners	**10%**	**23.0**	**14.0**	**6.0**
	25%	**37.0**	**17.0**	**7.0**
	50%	**56.0**	**20.0**	**9.0**
	Mean	**54.3**	**19.9**	**12.0**
	75%	**69.0**	**24.0**	**18.0**
	90%	**83.0**	**25.1**	**23.0**
MLAT Reach	**10%**	**605,283**	**56,419**	**316,215**
	25%	**640,807**	**66,918**	**414,541**
	50%	**685,620**	**102,179**	**476,768**
	Mean	**670,296**	**92,435**	**451,961**
	75%	**708,884**	**115,994**	**513,251**
	90%	**714,749**	**118,352**	**538,208**

Table 4. Statistics from 10,000 trials for pseudocountries generated by the constraint sets considered here.