Cryptographically secure detection of mirror worlds

Hristo Stoyanov

under the direction of
Professor Sharon Goldberg
Department of Computer Science
Boston University

Research Science Institute
July 30, 2014

Abstract

The Resource Public Key Infrastructure (RPKI) has been introduced as a way of authorizing Border Gateway Protocol (BGP) route announcements. The highly centralized structure of the RPKI provides security guarantees against external threats, e.g. prefix hijacking, but allows for the unilateral revocation of allocated resources. Recent efforts propose changes to the RPKI to create accountability of such unilateral actions. The project under consideration continues these efforts by providing a mechanism for ensuring global consistency. We solve the global consistency problem by constructing a k–connected graph containing all 2–party audits that the honest Autonomous Systems must perform to ensure that no mirror worlds exist.

Summary

Two computers on the Internet address each other by the so called IP addresses. Until recently, there was no verification that a specific computer is the right receiver of information. To address this security problem, a new highly centralized infrastructure is being created. However, this infrastructure gives power of a few organizations to unilaterally deny access to the Internet. As prevention proves hard, mechanisms for detection of this type of actions have been created. Continuing recent research efforts, we create an additional mechanism for ensuring that every network device has the same information about ownership of IP addresses. This mechanism proves efficient in decentralizing the propagation of this information.

1 Introduction

Cryptographic systems often require a trusted entity that is always consistent and honest. As such, this entity acts as the arbiter when disputes arise. In theory, this allows for the creation of highly robust and secure systems. Implementing this in practice proves difficult. Since the trusted entity is an organization of people, the entity might abuse the power it has.

An example of a system that takes this centralized approach is the recently introduced *Resource Public Key Infrastructure*. The purpose of this system is to create a mechanisms for verifying the owner of a set of virtual identities in the Internet. Physically, the Internet consists of a multitude of *Autonomous Systems* – independent and self-contained subnetworks of devices, and the connections between them. When devices from different *Autonomous Systems* communicate, they need a mechanisms for addressing each other. The virtual identities that devices assume are called IP addresses. The *Resource Public Key Infrastructure* serves as a public directory of which Autonomous Systems own a particular set of IP addresses. An *authority* (that is one of the trusted entities in the Resource Public Key Infrastructure) digitally signs the information in the public directory. All Autonomous Systems acquire that public directory and validates ownership against it.

The current design of the Resource Public Key Infrastructure allows the trusted entities to revoke ownership of IP addresses [1, 2]. By revoking the ownership of a set of IP addresses, the devices that have been using them are effectively isolated from any access to the Internet.

One way to achieve this is by executing the so called "mirror worlds" attack [2]. An authority can create these "mirror worlds" by presenting different misleading versions of the ownership directory. An example of a "mirror worlds" attack is presented in Figure 1. We have the following simplified setting:

- Trent is the authority that manages the public directory. Trent's goal is to prevent Bob from communicating with Carol.
- Alice and Bob want to communicate with Carol.
- Carol is in Chicago.
- Alice asks Trent where Carol is and the response is "Chicago".
- Bob asks Trent where Carol is and the response is "New York".
- Alice can successfully communicate with Carol.
- Bob tries to communicate with Carol, but fails.

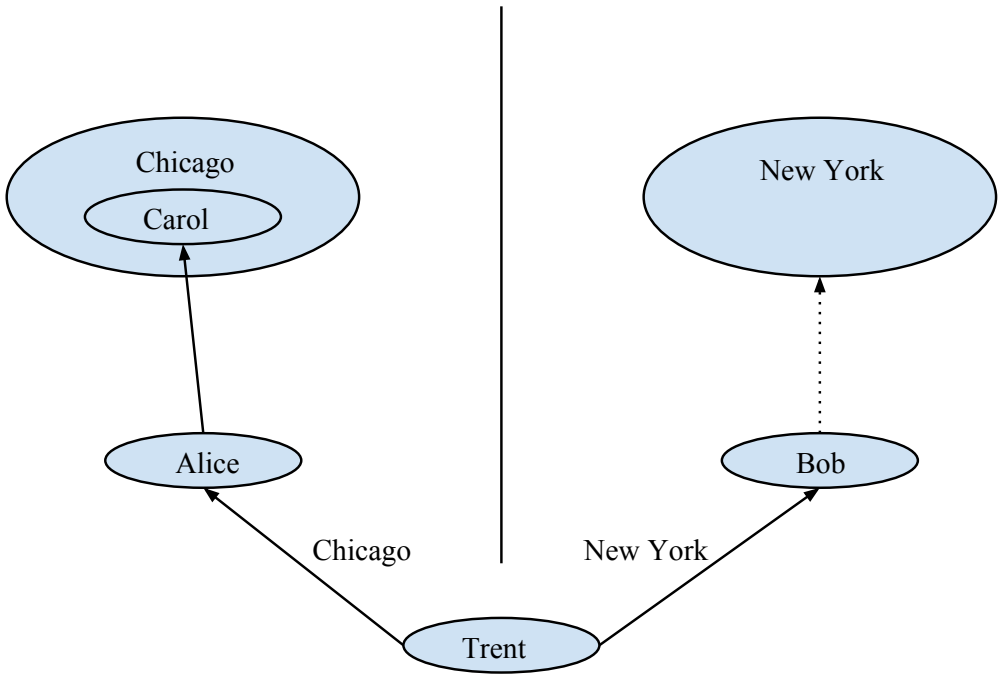

Figure 1: Alice and Bob live in "mirror worlds" [2] - they both have signed information about where Carol is, but the information is different.

Both Alice and Bob validate the information they receive but they cannot detect an inconsistency as locally the information is signed. However, Bob cannot reach Carol.

By managing this type of mirror worlds, an authority can selectively deny access to resources that it has allocated.

The Resource Public Key Infrastructure tries to secure one of the fundamental protocols of the Internet [3]. The protocol has mutliple issues [4, 5] that have led to severe [6, 7] attacks. Kuerbis and Mueller argue [8] that the introduction of the Resource Public Key Infrastructure "alters the distribution of power and economic benefits". For that reason, the Internet Service Providers (owning and operating almost all major connections and Autonomous Systems) "... show little inclination to adopt RPKI [Resource Public Key Infrastructure] en masse." Kuerbis and Mueller continue: "They [Internet Service Providers] are deeply concerned about the potential loss of autonomy...".

We refer to the detection of these "mirror worlds" as the *global consistency problem.*

Two parties can easily compare their information, however, when there are more than $50,000$ Autonomous Systems, ensuring that all of them live in a "single world" is not as trivial (*e.g.* the ASes might be split in large groups where each group lives in a different "world"). Additionally, some of the Autonomous Systems can support the separation by presenting different views of the public directory to other Autonomous Systems depending on which "world" they are from.

For a sufficiently small number of Autonomous Systems the problem has a trivial solution – everyone compares their information with everyone else. However, this is highly inefficient. For n Autonomous Systems, if n is sufficiently large, it requires $\binom{n}{2}$ comparisons.

The goal of the current project is to find an efficient solution of the global consistency problem. In order to do so, we represent the current infrastructure of the Internet as a graph with all Autonomous Systems as vertices and all connections (physical and logical) as edges. Assuming that there are less than k dishonest Autonomous Systems ($k \in \mathbb{N}$), we construct *k–connected subgraph* of the original graph. Before going further, let us introduce some standard definitions from graph theory:

Definition 1. A *graph* is the ordered pair $G = (V, E)$ comprised of a set of vertices V and a set of *edges* E. Each edge is defined by an unordered pair $(u, v) \in E$ such that $u, v \in V$.

Definition 2. A *complete graph* $G(V, E)$ is a graph for which $\forall u, v \in V$ and $u \neq v$ there is an edge $e \in E$ such that $e = (u, v)$.

Definition 3. A *subgraph* of a graph G is a graph $G'(V', E')$ such that $V' \subseteq V$ and $E' \subseteq E$.

Definition 4. A *connected* graph is a graph $G(V, E)$ such that there is a path between any two nodes $u, v \in V$. Conversely, a *disconnected* graph is a graph that is not connected.

Definition 5. A *k–connected graph* is a graph $G_k(V, E)$ for which k is the smallest number of vertices such that removing k vertices makes the graph disconnected.

Let V be the set of Autonomous Systems and E be the set of all connections (physical and logical) between them. Since all Autonomous Systems are connected, $G(V, E)$ is complete. Suppose that less than k of the Autonomous Systems are dishonest. We solve the global consistency problem by constructing a k–connected subgraph $G'(V, E')$ of $G(V, E)$ that represents all comparisons that Autonomous Systems need to perform in order to make sure that all of them have the same information, regardless of the dishonest Autonomous Systems. It is relatively easy to see that, if there are at most k dishonest Autonomous Systems, and if $G'(V, E')$ is k–connected, then there are no mirror worlds.

To construct G' we use a distributed algorithm implementing the Randomized Nearest Neighbor Scheme publish by Khan et al. [9]. The algorithm was originally designed for use in wireless sensor networks, but we find that it is applicable in the process of solving the global

consistency problem. The algorithm however assumes that no vertex in the graph will be intentionally malicious. We examine the worst case implications of such actions, construct a mechanism for their detection and prove its correctness as follows:

Theorem 1. Let $G_k(V, E)$ be a graph with the following property:

$$\forall v_i \in V : |\{(v_i, v_j) \in E | i < j\}| = min\{k, |V| - i\}$$

Let V_m with $|V_m| = p$ and $p < k$ be a set of malicious vertices. Then, the subgraph that contains only honest vertices $H(V - V_m, E')$ of G_k is $(k - p)$–connected.

Since all honest Autonomous Systems are part of a connected graph and due to the transitive property of the equality comparison, we conclude that all honest Autonomous Systems either:

- have proven that is no difference in the information acquired by the honest Autonomous Systems; or
- have detected the existing differences.

By constructing a provably secure mechanism for detecting mirror worlds, this paper contributes to the multiple ongoing research [2, 10] of improving the resilience of the Resource Public Key Infrastructure to internal threats.

In Section 2.1 we give more details on the inner workings of the process of establishing communication between devices part of different Autonomous Systems. In Section 2.2 we discuss the current approach for authorizing usage of IP addresses. An overview of recent proposals to improve the mechanism is presented in Section 2.3 to provide context for the current paper. In Section 3 we present a novel approach towards ensuring the consistency of information among all Autonomous Systems. Section 4 analyzes the efficiency of the solution and the espected resource usage.

2 Background

What follows is a more formal explanation of the mechanisms behind routing and the validation of information in the Resource Public Key Infrastructure work.

2.1 Routing

An *Autonomous Systems* (AS) is a logical entity that encompasses a collection of devices. An Autonomous System is independent and the internal operation and exchange of information within one system is left to its operator. Some pairs of ASes have a physical connection - a cable connecting a device from one AS to a device from another AS. *Routing* creates logical connectivity - a sequence of physically connected Autonomous Systems where each Autonomous System forwards information to the next one in the sequence.

Each Autonomous System is identified by a unique *Autonomous System Number* (ASN). As the Autonomous Systems and the physical connection between them have a dynamic nature (new ASes can be created, existing ones might become defunct, physical links can be built or fail) the sequences created for routing have to be robust against changes. Since the IP addresses act as an abstract over the physical nature of the Autonomous System Numbers, addressing a certain IP address is persistent, even if the physical infrastructure changes.

Several organizations have been created to coordinate the usage of IP addresses. These organizations are called Regional Internet Registries (RIRs) and each one has a designated geographical region of operation. The task of these registries is to ensure that only one AS uses a set of IP addresses. Once a registry allocates a certain set of IP address to an AS, that AS announces the association between its ASN and that specific set of IP addresses. The announcement is sent to the physical neighbors of that AS who propagate it further throughout the Internet.

These announcements are not validated and are trusted by default [2, 4]. This has led to multiple severe attacks [6, 7] that divert traffic from the original owner of a set of IP addresses to a malicious recepient [5]. A natural question arises - how do we verify if an announcement is by the rightful owner of a set of IP addresses?

2.2 Trust, but verify

The Resource Public Key Infrastructure (RPKI) has been created as a mechanism for authorization the ownership of a certain set of IP addresses [3]. This is achieved by creating a hierarchical structure of organizations who own a set of IP addresses and can suballocate a part of that set to other organizations. The Regional Internet Registries (RIRs) are at the top of the structure. They issue a *certificate* [11] by digitally signing an object that contains an Autonomous System Number, a public key and a set of IP addresses. The RIRs act as *trust anchors* – that is trust in their certificates is assumed (Figure 2).

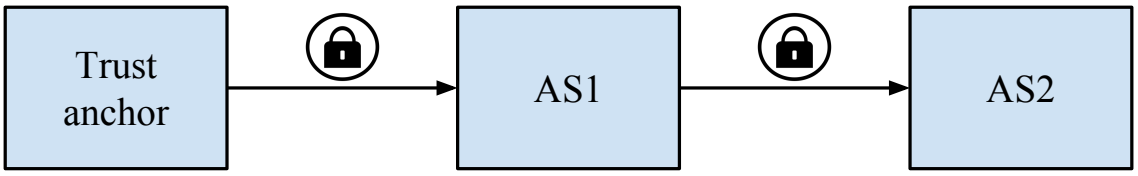

Figure 2: An example of a *chain of trust* – the trust in the certificate of AS2 is derived by the trust in the certificate of AS1, in turn derived by the trust anchor of RIR1.

The RPKI objects are several types of signed information. *Resource Certificates* (RC) are to be understood as defined by the X.509 standard [11] with the addition information of what set of IP addresses they are allowed to use. Other RPKI objects are *Route Origin Authorization* (ROA) and *manifests*. Certificates allow further suballocation of resources to other entities. The ROAs identify a specific Autonomous System as authorized to use a set of IP addresses. Manifests are described in more detail in Section 2.3.

The Resource Public Key Infrastructure is highly centralized (there are only five (5) registries in comparison with about 50000 Autonomous Systems). The current implementation of that system allows for the *unilateral revocation* of the right to use a set of IP addresses [1]. As shown by Heilman et al. [2] this gives the Regional Internet Registries powerful technical means to perform *take downs* of IP addresses, thus rendering the victim inaccessable through the Internet. These take downs can be done silently [2], as there is no detection mechanism. This stark departure from the current situation (where RIRs allocate IP addresses but have no influence on the addresses' further usage [2]) raises concerns in the rest of the involved organizations [8].

2.3 Record history

Heilman et al. [2] propose changes that increase the transparency of the actions of the authorities, introduce requirements of consent for revocation of resources and create mechanisms for auditing the published ownership information.

A *manifest* is a list of published and signed objects by an authority. Manifests become *normative* - objects are considered valid only if listed in the manifest. To publish information, an authority has to maintain a publishing point. The publishing point contains exactly one signed manifest. Objects there are identified by name and hash value [1] . The name provides information on how the object can be retrieved. The hash value of that object has multiple purposes. First, it acts as a checksum for a third party to assure herself that she has obtain the proper object. Second, by providing a hash value and signing this list, the authority commits to a specific version of the object. The object, the manifest and the signature are a proof that at that moment of time, the object was valid.

Every manifest that an authority issues need to contain the hash value of the previous

[1]Hash is the value of a one-way function H. It is computationally intractable to find $x \neq x'$ such that $H(x) = H(x')$. The hash value usually is of fixed size despite the size of the initial x. This assures us with a very high degree of certainty that if an object matches the intended hash value no tampering has occured.

one. By creating this *hash-chain* the authority provides anyone with the ability to audit and check all intertmediate states and track the changes that has happened to a certain object (e.g. certificate) in a provably secure way.

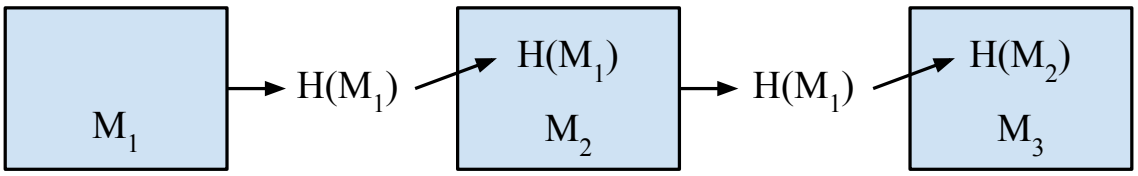

Figure 3: An example of *hash-chaining* - each message M_{i+1} containts the hash-value of M_i.

Manifests are dynamic as new authorizations are issued on a daily basis. Thus, manifests need to be comparatively short-lived. Certificates usually have a validity term of several years. Manifests should be usually valid for a few days.

2.4 A 2–party audit

Let us consider two Autonomous Systems - Alice and Bob, who would like to perform an audit of the information issued by an authority. Heilman et al. show [2] that Alice and Bob only need to be sure that the latest manifests they have acquired are the same. Since manifests are hash-chained, the equality between the latest manifests they have assures them that all the previous elements in the hash-chain match as well.

3 Build a k-connected graph

In this section we describe a novel approach towards solving the global-consistency problem. First, create a k–connected graph that ensures the connectivity of the honest auditors in the presence of at most $k-1$ malicious auditors. We apply a distributed implementation of a scheme for constructing this type of graph. We prove that the algorithm still results in

a connected subgraph of honest nodes despite that dishonest ones can distort the graph in their favor.

Let $G(V, E)$ be the complete graph with vertices all Autonomous Systems in the Internet and edges – all possible (physical or logical) links between them. Here, we explicitly assume that communication links will not be broken by the intermediaries. We will further discuss the practicality of this scheme in case this assumption does not hold in Section 4.

The 2–party audit (Section 2.4) assures the two auditors of having the same publication history. To ensure the same property in 3–party audit each party could execute the protocol with all the rest. Generalizing this for an n–party audit we find the required comparisons to be $\binom{n}{2}$, which becomes impractical as n grows.

To ensure the connectivity of honest Autonomous Systems in the presence of at most $k-1$ dishonest ASes, we need to build a k-connected subgraph G_k of the initial complete graph G that includes all ASes. No matter what the behavior of the dishonest ASes in that situation, the honest ASes are connected. By virtue of:

- the connectivity property of the graph G_k; and
- the transitive property of the equality comparison of manifests

follows that every honest auditor has acquired the same signed information by the authority.

3.1 Algorithm

To prove the connectivity of the honest Autonomous Systems, we will need the following result proven by Khan et al. [9]:

Proposition 1. Let $G = (V, E)$ be a graph on $V = \{v_1, v_2, \ldots, v_n\}$ with $n \geq k + 1$ so that every v_i has at least $\min\{k, n - i\}$ neighbors in $\{v_{i+1}, v_{i+2}, \ldots, v_n\}$. Then G is k–connected.

Khan et al. [9] describe a scheme, called the Nearest Neighbor Scheme, for constructing low-weight k–connected spanning subgraphs.

We start with a complete weighted graph $G(V, E)$ that contains all vertices and all possible edges between them. We then introduce a random total order of the vertices. Each vertex chooses a *rank* uniformly at random. We say that $u < v$ if and only if $rank(u) < rank(v)$. The ranks are chosen in a way that ensures there are no equal ranks with a sufficiently high probability. Once a total order $\{v_1, v_2, \ldots, v_n\}$ of the vertices has been introduced, each vertex v_i connects to at least $\min\{k, n-i\}$ vertices in the set $\{v_{i+1}, v_{i+2}, \ldots, v_n\}$. Each vertex chooses what connections to make by preferring edges with as small weights as possible. This minimizes the total weight of the resulting k–connected spanning graph. In our use of the algorithm we choose each edge to have unit weight. Figure 4 (Appendix A) describes the implementation of the Random Nearest Neighbor scheme shown by Khan et al.

3.2 Honest–nodes connectivity

Khan et al. explicitly state that the ranks they choose remain unchanged throughout the execution of the algorithm [9]. However, we use the algorithm in a setting where behavior of each vertex can be intentionally malicious. To prove correctness of the algorithm in that setting, we need to prove that the honest Autonomous Systems remain connected regardless of the malicious behavior. Our main result is the following theorem:

Theorem 1. Let $G_k(V, E)$ be a graph with the following property:

$$\forall v_i \in V : |\{(v_i, v_j) \in E | i < j\}| = min\{k, |V| - i\}$$

Let V_m with $|V_m| = p$ and $p < k$ be a set of malicious vertices. Then, the subgraph that contains only honest vertices $H(V - V_m, E')$ of G_k is $(k-p)$–connected.

Proof. Let us denote the set of honest nodes by V_h (thus, $V_h = V - V_m$ and $|V_h| = n - p$). The connections each honest node $v_i \in V$ creates are $\min\{k, n-i\}$. Since there are p malicious

nodes, the number of connections to higher-ranking honest nodes is $\max\{0, \min\{k, n-i\}-p\}$ for each honest node v_i. There is a corresponding node $q_s \in V_h$ for each honest node $v_i \in V$. Similarly, for each edge $(v_i, v_j) \in E$ (where $i < j$) there is corresponding edge $(q_s, q_t) \in E'$. Since V is a totally ordered set, and $V_h \subseteq V$, the ordering on V induces the same ordering on V_h. Hence, every $q_s \in V_h$ has at least $\max\{0, \min\{k, n-s\}-p\}$ connections to higher-ranking honest nodes. Since $1 \leq s \leq n-p$, this implies that every q_s has at least $\min\{k-p, (n-p)-s\}$ neighbors in the set $\{q_{s+1}, q_{s+2}, \ldots, q_{n-p}\}$. Directly applying Proposition 1, we conclude that $H(V_h, E')$ is $(k-p)$–connected.

Hence, it is straightforward that the subgraph that contains all honest Autonomous Systems is connected.

3.3 Security guarantees

In the described threat model, we proved that the scheme is k–secure in the n–party audit setting. The implementation works for every $k \leq n-1$ (the case where $k = n-1$ is a full graph). Deriving a practical value of k is out of the scope of this paper. Choosing a specific value requires community consensus and discussion that should take place at the appropriate forums.

Consider the Sybil attack [12] where in one organization operates multiple identities to take control of a distributed system. To avoid the possibility of impersonation, the communications between any two nodes should be encrypted. Since the Autonomous Systems are both the subjects of certification and the auditors of the certifier, one possibility would be using the public keys associated with these certificates. However, this allows the certificate authority to impersonate auditors. Hence, we propose the usage of separate SSL certificates.

4 Analyze

Section 4.1 gives estimates on the resources that the system requires both in terms of time to construct the graph and the traffic needed to coordinate the building of the graph. In Section 4.2 we present a procedure for analyzing the physical layer of the Internet. By doing so, we can get rough estimates on what practical values of k should be considered.

4.1 Consider practicality

Khan et al. [9] prove that the complexity of the algorithm is $\Theta(\lg \frac{n}{k})$ and the expected number of messages is $\Theta(kn \lg \frac{n}{k})$. This assessment is with the assumption that a node can send messages to several nodes simultaneously.

We argue that this is a realistic assumption in the current state of the Internet infrastructure. We will perform a back-of-the-envelope estimation of the size of the created traffic and the time required to transmit it. One FIND message (more in Appendix A) has the combined size of one ASN, one rank and a signature. In this case id is the Autonomous System Number which is 4 bytes [13]. We estimate the size of the rank to be 32 bytes (thus, the rank can be chosen in the space of 2^{256}). A pessimistic estimate for the size of the signature is $4KB$ [11]. Even if we assume that encoding and transmission require 10 times the initial message size, the final estimate for a single message with all the overhead is $40KB$.

Assume that the amount of ASes will double to $1,000,000$. This means that the total amount of traffic for one node that has to probe all other nodes is $1,000,000 * 40KB = 40GB$. Since the smallest link between two ASes is on the order of 100Mbps+, the longest time required for sending $40GB$ of traffic is $40GB/100Mbps = 320,000Mb/100Mbps = 3200seconds$. This is rougly 54 minutes, even though we made the worst case assumptions. As the RPKI should be updated once a day [3], an hour for the construction of the graph is relatively efficient.

4.2 Examine the infrastructure

We assume previously that no Autonomous Systems will break connections between other ASes (*e.g.* by blocking traffic instead of routing it). Here, we discuss how to analyze a similar scheme: instead of creating a k–connected overlay, everyone executes the 2–party audit with the physical neighbors they have. This will form several subgraphs that have various degrees of connectivity.

We pose the following question: *given the graph of physical connections between ASes and a parameter k, what are the k–connected subgraphs with the maximum number of vertices?* We can apply several algorithm to analyse the graph of ASes. First, we find subgraphs that have a *minimal degree* of at least k. For each such subgraph we test for k–connectivity.

4.2.1 k–cores

The *minimum degree* of a graph G is denoted by $\delta(G)$. The minimum degree is a standard upper bound for the connectivity of a graph. Proof can be given by a simple contradiction. Suppose a graph $G(V, E)$ is k–connected. There is a node u that defines the minimum degree of the graph $\delta(G) = \deg(u)$. Assume that $k > \delta(G)$. Yet, by deleting all neighbors of u, we disconnect u from the graph. This is a contradiction with the definition of a k–connected graph as the number of neighbors $k > \delta(G)$, thus $k \leq \delta(G)$.

If H is a maximal connected subgraph of G with $\delta(H) \geq k$, we say that H is a *k–core* of G. The notion of *k–core* introduced by Seidman [14] enables the finding of a subgraph with high minimal degree, but this subgraph may not necesseraly be k–connected [15]. Finding all k–cores is achieved by a simple algorithm [16]: recursively delete all vertices v_i and all edges (v_i, q) that have degree $\deg(v_i) < k$ in the remaining graph G'. This results in either a set of the k–cores of the initial graph G or, if there are no k–cores, an empty set.

4.2.2 Calculate connectivity

Menger proves [17] that a graph is k–connected if and only if every pair of vertices is joined by at least k *vertex*-disjoint paths [15]. Network flow algorithms can be applied to calculate the number of *vertex*-disjoint paths between two nodes. A flow of k between s and t implies k *edge*-disjoint paths.

There is a construction that allows us to create a *directed* graph $G'(V', E')$ where each *edge*-disjoint path is equivalent to a *vertex*-disjoint path in the *undirected* graph $G(V, E)$ [15]. First, for every node s in G we add two nodes, s_{in} and s_{out}, and a directed edge (s_{in}, s_{out}) in G'. For each edge $(s, t) \in E$ we add two edges, (s_{out}, t_{in}) and (t_{out}, s_{in}), to E'. The number of *vertex*-independent paths from s to t in G is equal to the number of *edge*-independent paths from s_{out} to t_{in}.

We use the Ford-Fulkerson algorithm [18] to calculate the number of edge-independent paths between two nodes. By applying this to every possible pair in the graph G and calculating the smallest such number we get the connectivity of the graph G.

4.2.3 Analyze of the AS-level graph

We use data accumulated by the Internet Research Lab at UCLA from several sources [19]. Monthly datasets list the connections and the number of days each connection has been observed during that month.

Figure 5 shows how the Autonomous Systems graph evolved in three consecutive years. The statistics are for June in 2012, 2013 and 2014. Unless stated otherwise, we consider only relations that have been observed for more than 10 days for that particular month. We can observe that the maximum k that still yiels a k–connected graph grows in the years. It is respectively 63, 74 and 77.

In Figure 6 we compare how big are the graphs if we consider only links observed more than 10 times during that month (June 2014) and the graphs what have all links observed

during that same month. There is a significant difference between the results we get for these two situtations (for example the 56–connected graph increases from 418 to 631 nodes, an increase of more than 50%). Since data collected by the UCLA's Internet Research Laboratory is incomplete [19], we argue that acquiring more data on the AS–level topology of the Internet will suggest even larger subgraphs for all values of k.

5 Conclusion

We define and examine the *global consistency problem* in the context of the Internet with the goal of creating a mechanism for safeguarding against malicious behavior from the *authorities* and at most $k-1$ dishonest Autonomous Systems. We design a scheme that applies an algorithm for building a k–connected spanning graph and next performs a 2–party audit for every edge of that graph. We first study the system, assuming that all Autonomous Systems are logically connected. We analyze the setting where the assumption of logical connectivity does not hold. The procedure and results of this analysis provides an insight on choosing a practical value for k to be considered by the relevant organizations. Furthermore, we prove that either the information is globally consistent or there exist local discrepancies which are detected by the honest Autonomous Systems. This additional mechanism contributes to the resilience of the Resource Public Key Infrastructure to internal threats.

6 Acknowledgments

I would like to thank my mentor Professor Sharon Goldberg from BU for the guidance and support throughout our work together. I would also like to express my gratitude towards the BU students who listened to my flow of random ideas and discussed them in detail - Ethan Heilman, Alison Kendlar and Aanchal Malhotra. I thank my tutor Dr. Jenny Sendova for

the restless reviews of my work. I thank Kati Velcheva, Adam Sealfon and Veselin Kulev for the edits and recommendations they gave me while writing my paper, and Linda Westrick for adding the finishing strokes. I express my deepest gratitude to the "America for Bulgaria" Foundation, "St. Cyril and St. Methodious International Foundation", the "Evrika Foundation" and the "American Foundation for Bulgaria" for sponsoring my participation in RSI, as well as to the CEE, the staff of the RSI and the MIT for my work would have been impossible without their efforts and support.

References

[1] D. Cooper, E. Heilman, K. Brogle, L. Reyzin, and S. Goldberg. On the risk of misbehaving rpki authorities. November 2013.

[2] E. Heilman, D. Cooper, L. Reyzin, and S. Goldberg. From the consent of the routed: Improving the transparency of the rpki. August 2014.

[3] M. Lepinski and S. Kent. RFC6480: An Infrastructure to Support Secure Internet Routing. Technical report, Internet Engineering Task Force (IETF), 2012.

[4] K. Butler, T. Farley, P. McDaniel, and J. Rexford. A survey of bgp security issues and solutions. *Proceedings of the IEEE*, 98(1):100–122, Jan 2010.

[5] S. Murphy. RFC4272: BGP Security Vulnerabilities Analysis. Technical report, Internet Engineering Task Force (IETF), 2006.

[6] A. de Beaupre. ISC Diary: BGP multiple banking addresses hijacked. 2013.

[7] Rensys blog. Pakistan hijacks YouTube. 2008.

[8] B. KUERBIS and M. L. MUELLER. Negotiating a New Governance Hierarchy: An Analysis of the Conflicting Incentives to Secure Internet Routing. *Communications & Strategies*, 1(81):125–142, 1st quart 2011.

[9] M. Khan, G. Pandurangan, and V. A. Kumar. A simple randomized scheme for constructing low-weight k-connected spanning subgraphs with applications to distributed algorithms. *Theoretical Computer Science*, 385(13):101 – 114, 2007.

[10] S. Kent and D. Mandelberg. Suspenders: A Fail-safe Mechanism for the RPKI (Draft). Technical report, Internet Engineering Task Force (IETF), 2014.

[11] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. RFC5280:Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. Technical report, Internet Engineering Task Force (IETF), 2008.

[12] J. R. Douceur. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 251–260, London, UK, UK, 2002. Springer-Verlag.

[13] Q. Vohra and E. Chen. RFC 6793: BGP Support for Four-Octet Autonomous System (AS) Number Space. Technical report, Internet Engineering Task Force (IETF), 2012.

[14] S. B. Seidman. Network structure and minimum degree. *Social Networks*, 5(3):269287, 1983.

[15] S. S. Skiena. *The Algorithm Design Manual*. Springer-Verlag New York, Inc., New York, NY, USA, 1998.

[16] A. E. Saríyüce, B. Gedik, G. Jacques-Silva, K.-L. Wu, and U. V. Çatalyürek. Streaming algorithms for k-core decomposition. *Proc. VLDB Endow.*, 6(6):433–444, Apr. 2013.

[17] K. Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 10(1):96–115, 1927.

[18] L. R. Ford and D. R. Fulkerson. Maximal Flow through a Network. *Canadian Journal of Mathematics*, 8:399–404.

[19] *Internet AS-level Topology Archive*. Available at http://irl.cs.ucla.edu/topology/.

A Algorithm

Figure 4 describes a distributed implementation of the Randomized Nearest Neighbor Scheme as shown by Khan et al. [9]. The algorithm uses the following notations:

- $\eta(u)$- the set of nodes to whom u is connected;
- $\Gamma_u(k)$ - the set of k nearest neighbors of u in the complete graph $\mathbb{K}_n$;
- FIND message - includes the $id(u)$ of the sender and the rank $r(u)$ the sender has chosen.

Distrubuted k–connected graph algorithm

Input: A complete graph $\mathbb{K} = G(V, E)$. We assume each node has a unique id from a totally ordered set.

Output: A k–connected subgraph G_k. On termination, each node knows which of its adjacent edges are in G_k.
Each node $u \in V$ executes the following protocol independently and simultaneously:

1. Choose the rank $r(u)$ as follows: generate a random number $p(u) \in [0, 1]$. We say $r(v) > r(u)$ if and only if $[p(v) > p(u)]$ or $[p(v) = p(u)$ and $id(v) > id(u)]$.

2. Find $|\eta(U)|$ nearest nodes q with $r(q) > r(u)$, and add the edges (u, q) to G_k. Find the q's as follows:

   $t \leftarrow 1$;
   repeat
   if $t = 1$ **then**
   u sends FIND messages to all $v \in \Gamma_u(k)$ simultaneously;
   end
   if $t \geq 2$ **then**
   u sends FIND messages to all $v \in [\Gamma_u(2^{t-1}k) - \Gamma_u(2^{t-2}k)]$ simultaneously;
   end
   until *u received k ACCEPT messages or probed all of its neighbors* ;
   $t \leftarrow t + 1$;

3. Upon receipt of a FIND message from any v, send back an ACCEPT message to v *if and only if* $r(u) > r(v)$.

Figure 4: Implementation of the scheme described by Khan et al. [9].

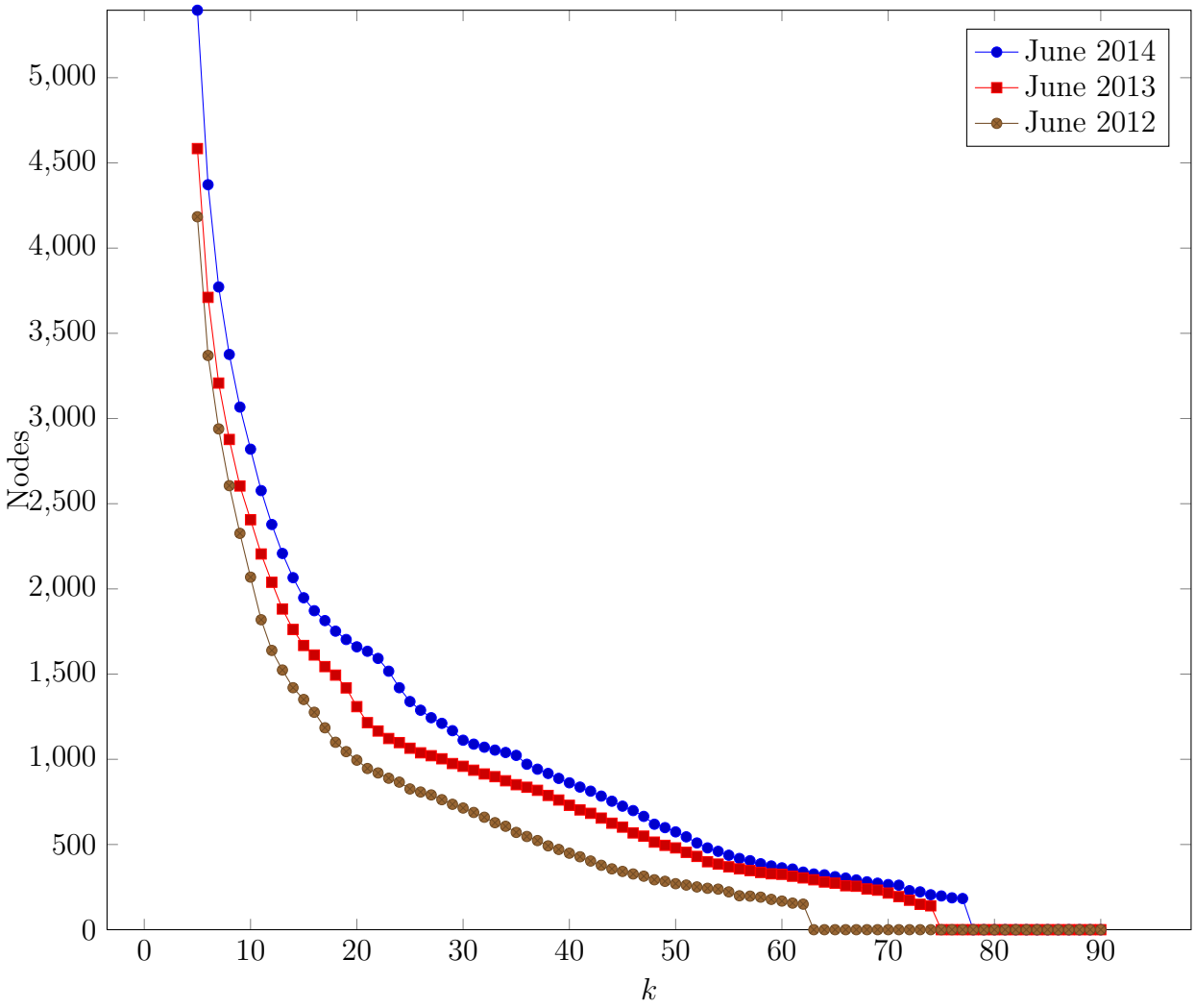

Figure 5: Number of Autonomous Systems that are part of a k–connected subgraph of the Internet. The graph is generated by links observed at least 10 times during June 2014. Data points are for $k \in [5, 90]$.

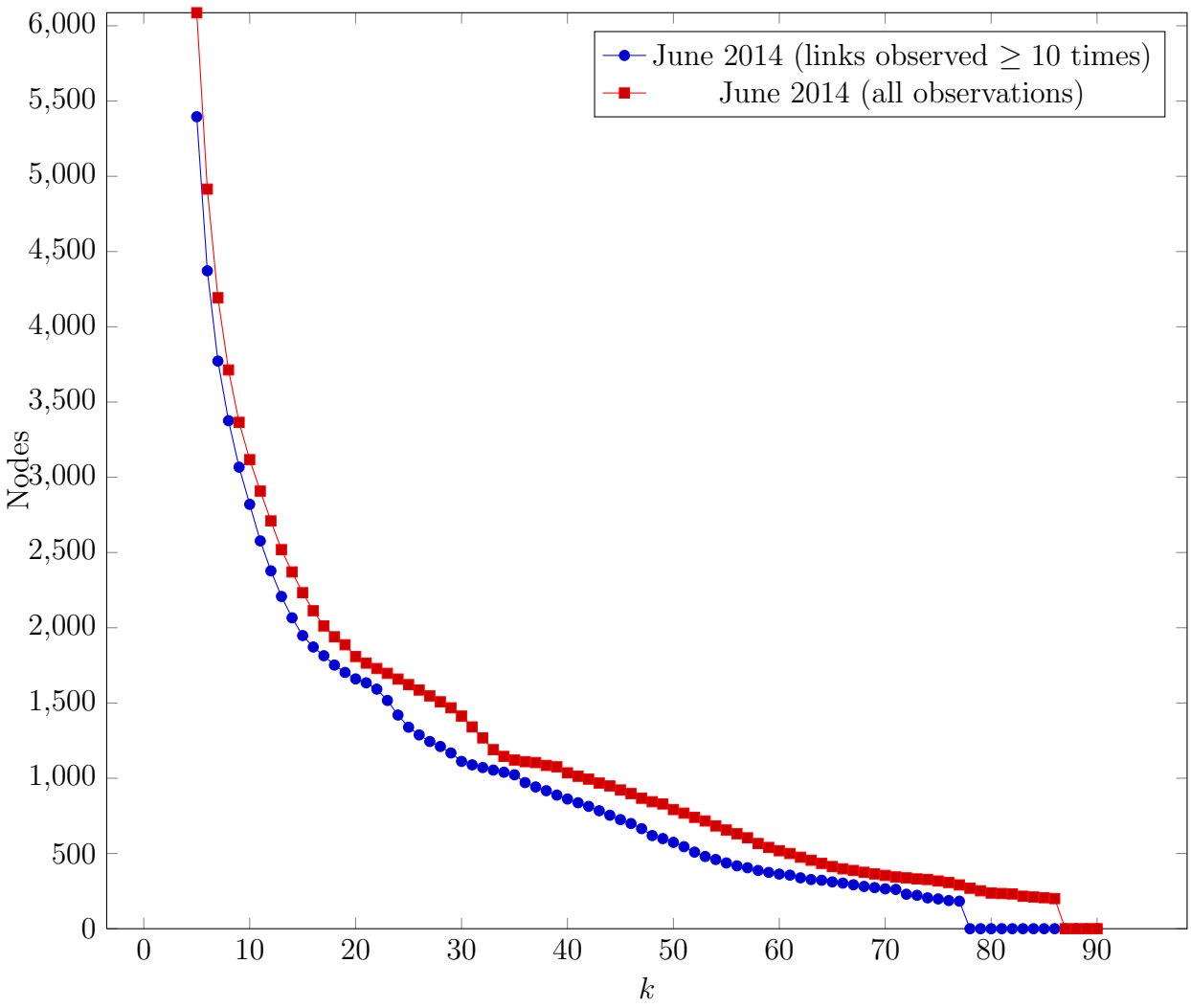

Figure 6: Number of Autonomous Systems that are part of a k–connected subgraph of the Internet. Data points are for $k \in [5, 90]$.