Verifying Keys through Publicity and Communities of Trust: Quantifying Off-Axis Corroboration

Eric Osterweil, Dan Massey, *Senior Member*, *IEEE*, Danny McPherson, and Lixia Zhang, *Fellow*, *IEEE*

Abstract—The DNS Security Extensions (DNSSEC) arguably make DNS the first core Internet system to be protected using public key cryptography. The success of DNSSEC not only protects the DNS, but has generated interest in using this secured global database for new services such as those proposed by the IETF DANE working group. However, continued success is only possible if several important operational issues can be addressed. For example, `.gov` and `.arpa` have already suffered misconfigurations where DNS continued to function properly, but DNSSEC failed (thus, orphaning their entire subtrees in DNSSEC). Internet-scale verification systems must tolerate this type of chaos, but what kind of verification can one derive for systems with dynamism like this? In this paper, we propose to achieve robust verification with a new theoretical model, called *Public Data*, which treats operational deployments as *Communities of Trust (CoTs)* and makes them the verification substrate. Using a realization of the above idea, called *Vantages*, we quantitatively show that using a reasonable DNSSEC deployment model and a typical choice of a CoT, an adversary would need to be able to have visibility into and perform on-path Man-in-the-Middle (MitM) attacks on arbitrary traffic into and out of up to 90 percent of the all of the Autonomous Systems (ASes) in the Internet before having even a 10 percent chance of spoofing a DNSKEY. Further, our limited deployment of Vantages has outperformed the verifiability of DNSSEC and has properly validated its data up to 99.5 percent of the time.

Index Terms—DNSSEC, DNDKEY, verification, p2p

1 Introduction

The DNS [14] has been one of the Internet's core infrastructure systems for almost 30 years. Now, with the DNS Security Extensions (DNSSEC) [4], [6], [5], DNS is becoming the first operationally deployed Internet-scale distributed system to be protected using public key cryptography. As of June 2011, the DNS root and major top level domains (e.g., .com, .gov) have been signed. Measurement data from SecSpider [2], [17] show that the number of signed zones roughly quadrupled between 2010 and 2011, from roughly 7,000 to over 30,000 (and was several hundred thousand at the time of this writing). As a robust and now potentially secured global database, there is a growing interest in using DNS as a general Internet-scale infrastructure to verify and bootstrap secure transactions. In particular, the IETF's DANE working group [7] proposes to use the "DNS to provide source authentication for public keys." This operational deployment combined with the work to add new services is an important watershed event that reflects an increased awareness that operators are gaining about securing core protocols and the potential to add new services.

In theory, DNSSEC is a simple matter of overlaying basic public key cryptography onto the existing DNS tree. In practice, however, DNSSEC (as designed) lacks the robustness property found in the original DNS design. DNS has thrived because its design tolerates failures and misconfigurations. DNSSEC is much less tolerant of misconfigurations and imperfect operations have already led to major outages for DNSSEC. For example, top level domains like `.arpa`, `.gov`, and `.fr` suffered outages from configuration errors that made their entire subtrees be unverifiable [10], [18], [15], [11]. We believe that DNSSEC's key verification design is technically "correct"; however, it needs a fundamental *enhancement* (not replacement) so that operational errors do not lead to wide-scale outages of subtrees. We should not consider it acceptable to say (for example) that all zones under `.gov` are unverifiable because of an operational misstep higher in the hierarchy.

An underlying challenge is that trust in a zone's keys is learned from a DNS hierarchy that was designed to distribute authority and provide name uniqueness, not provide key verification. Is there a fundamental misalignment in using DNS' hierarchy for key verification? Further, how do we define the "valid keys" for a zone? Are they the keys on one name server, the keys an operator just generated, the ones pointed to by another company (the

- *E. Osterweil and D. McPherson are with the Verisign Labs, 12061 Bluemont Way, Reston, VA 20190. E-mail: {eosterweil, dmcpherson}@verisign.com.*
- *D. Massey is with the Computer Science Department, Colorado State University, 1873 Campus Delivery, Fort Collins, CO 80523-1873. E-mail: massey@cs.colostate.edu.*
- *L. Zhang is with the Computer Science Department, UCLA, 4732 Boelter Hall, Los Angeles, CA 90095. E-mail: lixia@cs.ucla.edu.*

Manuscript received 16 Sept. 2012; revised 12 Apr. 2013; accepted 17 June 2013; published online 28 June 2013.
Recommended for acceptance by X. Li, P. McDaniel, R. Poovendran, and G. Wang.
For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number TPDSSI-2012-09-0873.
Digital Object Identifier no. 10.1109/TPDS.2013.168.

parent zone), the ones on an HSM, or on a thumb drive? And how does one distinguish between a key rollover, a misconfiguration, or an attack? A verification system must tolerate the Internet's chaotic setting, but what kind of verification can one derive for an Internet-scale system with dynamism like this?

In this paper, we propose to address this problem with a new model, called *Public Data*, that treats operational deployment specifics as the verification substrate. That is, cryptography is currently used as the core substrate (or foundational building block) of verification in DNSSEC today. All protections are built upon cryptography verification. Our intention is to use operational deployment specifics as foundational elements on which to build an orthogonal verification scheme to DNSSEC's current cryptographic design. Starting from the observations that *redundancy can overcome errors*, *publicity increases verifiability*, and *who to trust is subjective*, the Public Data model (based on [16]) has each resolver operator cull observations from their own choice of witnesses that are located at different *vantage points* in the network. These witnesses become a resolver selected *Community of Trust (CoT)*, which capitalizes on observations of data taken from multiple data sources, from different vantage points, at different times, and collected through different network paths. Further, Public Data's verification framework allows zone operators to augment their redundancy in a new way by serving their keys over different protocols and on different servers. Verification can be aided by the *redundancy* of serving keys from different sites and over different protocols (like to a number of secondary name servers and websites). Thus, misconfigurations at one site, or over one protocol (such as in a zone file) do not necessarily result in outages. We use this framework to codify the common sense notion that crosschecking enhances confidence so that real systems can be designed using these primitives, and we use this model to *quantitatively* demonstrate the efficacy of this technique.

We have implemented a candidate Public Data system called *Vantages*, and have been running it for over two years in our labs. We show that while it has *already* outperformed the hierarchical approach, it can also be used to *compliment* hierarchical verification. In fact, we were able to properly validate its data up to 99.5 percent of the time in our specific deployment. We show that under a reasonable DNSSEC deployment and choice of CoT, an adversary would need to be able to perform on-path Man-in-the-Middle (MitM) attacks on arbitrary traffic of up to 90 percent of all of the Autonomous Systems (ASes) in the Internet for even a 10 percent chance of spoofing a DNSKEY.

The remainder of this paper is organized as follows: We provide an overview of the operational status of DNSSEC and outline key challenges that any verification system must face in supplemental material (available at http://ieeexplore.ieee.org). Then, Section 2 presents the Public Data model. Section 3 applies these concepts to build the Vantages system. Sections 4 and 5 provide security analysis, deployment results, and evaluation, respectively, before concluding in Section 6.[1]

1. We discuss many additional details (including related work) in supplemental material, available online.

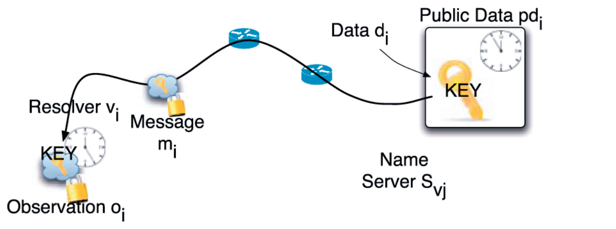


Fig. 1. An observer at v_i queries Data Source S_{v_j}, producing Observation o_i or datum pd_i.

2 Public Data

The *Public Data* theoretical model demonstrates that rather than using Boolean assurances of a crypto-enhanced design alone, one can incorporate operational nuances and data source redundancy as fist-class elements of DNSSEC's crypto key verification. That is, cryptography is currently used as the core foundational building block (or substrate) of verification in DNSSEC (and other PKI models) today. All protections in these types of systems are built upon cryptographic verification substrate. Our intention is to use operational deployment specifics as a different kind of foundational element on which to build an orthogonal verification scheme to augment DNSSEC's current cryptographic design. The model's basic intuition is that widely known Public Data become quantifiably more difficult to spoof as its public presence increases. Using Public Data, Alice (an operator) can compare her own observation of a DNSKEY with observations witnessed at other topologically diverse locations. She relies on the topological diversity of her witnesses to protect her against the attacks we discuss in Section 4.1. In essence, the network's topological path diversity becomes the mechanism that Eve must defeat.

2.1 Model

All Public Data are requested from, served from, and transferred through *Vantages*. We model the Internet as a set of network Vantages whose connectivity to each other can be expressed as the graph $G = (V, E)$. A vantage $v_i \in V$ is essentially a network node at an IP address. Examples of Vantages include DNS resolvers, name servers, web servers, routers, and so on.

In our model, DNSKEYs are served from *Data Sources* (name servers, web servers, etc.). A server is located at some vantage and is responsible for serving a zone's data. Fig. 1 shows a Data Source at vantage v_j, denoted S_{v_j}. When a datum d_i is served from a Data Source S_{v_j}, it becomes an immutable Public Datum, $pd_i = (d_i, t_k)$, where d_i is the datum itself and t_k is the inception time of d_i. A *Public Data Source* $S_{v_j} = \{pd_0, \ldots, pd_m\}$ is characterized by its network vantage of v_j and all of the Public Data it has ever served.

Data are exchanged using a message $m = (d_i, Sig_K(d_i))$, where d_i is an opaque data item, $Sig_K(d_i)$ is a signature that covers d_i, and the signature can be verified by the crypto key K. A message m_i sent from v_j to v_k will traverse a single acyclic path of Vantages (determined by a routing protocol) denoted: $\sigma_{(j,k)} = (v_j, \ldots, v_k)$. When vantage v_j receives the message, it creates an observation: $o_i = (v_j, m_k, t_l)$ that contains the vantage that made it, the message, and the time at which it was received. Furthermore, if the recipient has already learned the verifying key K, then the message's authenticity can be checked. That is, this $Sig()$ can be used to verify m. Later, we will use this to secure the message exchanges between Vantages who are peered and have learned each others' keys. Verifying the key is discussed further in Section 2.2.

Finally, we also assume that even if a server S removes one piece of data d from its currently served set, it must be possible to prove that S served d in the past (a weak form of nonrepudiation). This mechanism is an important part of verifying Public Data as it helps protect client resolvers from malicious servers.

2.2 Verification Processes

In our model, validity is a notion of the true authenticity of data. To define validity, we introduce a message oracle Θ that is always able to determine validity with 100 percent certainty.[2] For a given datum d_i and time t_i, $valid(\Theta, d_i, t_i) = true$ iff 1) d_i came from the source it reported, 2) the time stamp t_i does not occur *before* the data were actually created at the source, and 3) d_i was still being served by the data source at t_i (i.e., it is not a replay of older data). Thus, $valid()$ only determines if an observation properly reflects data from a Public Data Source (this is not "ground truth" in the sense of whether the data's meaning holds any real-world significance). Based on this, we can define an attack as $valid(\Theta, d_i, t_i) = false$.

In Public Data, *validity* and *verifiability* are distinct concepts. Since clients do not have access to Θ (and thus cannot assess validity), they are instead concerned with the *verifiability* of data. A resolver at v_i creates a query message m_i at time t_0 and sends it to a name server S_{v_j} over the path $\sigma_{(i,j)}$ (determined by routing at the network layer). When the server S_{v_j} receives this query, it constructs a response message m_j (containing the data d_k from its current pd_k), and sends it back over the reverse path $\sigma_{(j,i)}$.

Our definition of *verifying* a data item d_i casts verification as a function of a continuous metric that uses a set of observations O and a user-defined threshold value p. The observations O are used to perform consistency checking of the data items. Each observation $o_i \in O$ could (for instance) be an observation made from Alice's vantage v_i to a specific name server $S_{v_j} \in V_Z$.

As one adds more observations from different Vantages into O, there is an increased chance that this will add paths that do not intersect with those already in O. Therefore, adding *independent* paths increases the number of Vantages that Eve must subvert to keep $verify()$ from converging on the valid answer.[3] Thus, we define a *Community of Trust* V_{CoT} as a set of Vantages (called *witnesses*) from which Alice can cull additional observations. The idea is to let Alice leverage her own judgment of real-world trust in other operators or organizations to *manually* select who she trusts to help her add path diversity to her O. In this model, we require Alice to have obtained the public keys for these witnesses through an out of band mechanism (to verify messages from them). By bootstrapping the keys from her V_{CoT} ahead of time, Alice can verify the crypto signatures on communications with her witnesses.

When Alice queries her V_{CoT}, and some observations report different values, any datum with greater than a p majority is chosen as the verified datum. In other words, p is the proportion of nodes that must agree to call a data item "verified." The specific algorithm for $verify(O, p)$ is described in supplementary information, available online.

As zone administrators increase their set of name servers V_Z and an operator increases his V_{CoT}, Eve must spend more and more to try and discover and compromise Vantages (V_e) that can intercept response messages. Essentially, Eve must try to be in the right place at the right time, and a well-provisioned deployment makes this systemically unrealistic. In Section 5.1, we will show just how effective the Internet's topology is at providing protections with this model.

3 Vantages

To demonstrate the effectiveness of the Public Data model, we have implemented it in an open source system called Vantages. Vantages is written in C++, it uses a SQLite back end, and it has been publicly available since early in 2009 [3]. CoTs are peer-to-peer networks, where each witness is an individual `vantaged` daemon. These daemons are designed to be installed alongside recursive resolvers (on the same hosts) so that they can use `libpcap` to automatically learn which zones to monitor. The daemons communicate over HTTP with peers in its CoTs. Finally, when setting up an instance, the operator must specify a PGP [25] key that is used to sign all data observed by the daemon. This allows Vantage daemons to secure the data messages they exchange. Additional details about its implementation are included as supplemental information, available online.

3.1 Evolving the Theory into Practice

"In theory, there is no difference between theory and practice, but in practice there is." Indeed [23], some of the operational complexities that Vantages faces require modifications from to the pure Public Data model, and some of them actually help bolster its security.

Mapping from Public Data to Vantages. The CoT concept in Public Data generically describes how a set of multiple Vantages can act as *witnesses* and cooperate to verify Public Data. Just as in Public Data, the Vantages system provides increasing protection as CoTs grow (as seen in Fig. 2). However, in a real system, not all parties will necessarily want to trust each other in the same CoT. The act of automatically discovering the *existence* of a node (such as by using a DHT [21]) should *not* be coupled with the process of bootstrapping trust and creating a CoT because their objectives are orthogonal. Trust (in the Internet) is not transitive and while any given user may elect to trust witnesses he or she knows, it is unreasonable to assume that she will necessarily want to trust the witnesses that other members of their CoT trust. We presume that operators should *manually* create their CoT, and not seek an autodiscovery mechanism. Rather, note the existing practice that operators follow of using trusted community forums

2. It is important to note that Θ is not accessible to any $v_i \in V$ and is only defined to disambiguate if $d_i \in pd_i$ or $d_i \notin pd_i$.

3. We note that adding observations to O does not necessarily add path independence, and we address this in Section 4.3.

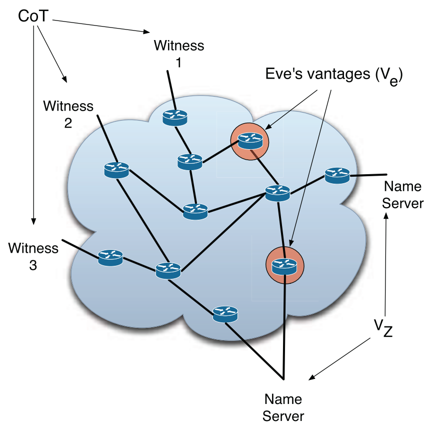


Fig. 2. If a CoT is composed of witnesses 1-3 (on the left), and Eve has compromised the three circled autonomous systems, at least one member of the CoT will see genuine values.

like NANOG's [1] regular face-to-face meetings meet and establish trust communities. We imagine the trust relationships of CoTs will resemble Fig. 3.

One notable benefit of this approach is that if a CoT gets fooled, it will *not* necessarily affect other CoTs because their independence also serves as a way to isolate failures. Shifting the onus of forming a CoT onto operators clearly sounds daunting. However, we address this in the following ways: 1) DNS best-common-practices already tell name server operators to spread their DNS secondaries out to other networks, and many operators (at universities and other organizations) already host secondary services for each other (so there should already be existing relationships to leverage), and 2) we introduce the concept of the Super-Sized Witness below.

Super-Sized Witnesses. In addition, some witnesses (such as large ISPs or distributed monitoring systems) may have points-of-presence at multiple distinct topological locations. These witnesses can make multiple observations from different locations, and share them within their CoTs, which then augments the system's overall verifiability. All verification operations keep the *data provenance* of how a decision was made. If a witness misbehaves, the operators in the CoT can see it, and choose to evict that witness. We will show that examples of open Super-Sized witnesses already exist, and we evaluate Vantages' deployment security analysis using measurements from one of these witnesses' actual deployment.

Aligning Costs with Benefits. The motivation to deploy a Vantage daemon is that an operator gains the benefit of the CoT they peer with, while their participation in that (or any other) CoT makes *its* verification stronger too. This is because its daemons are intended to be deployed alongside, and directly interface with, each of an operators DNS resolvers. Thus, Vantages' trust model properly aligns costs and benefits: operators deploy as many as benefit them, and peer with whomever they have amicable relations, and the more they invest in their deployment, the better their protections are. This is one core differentiator between Vantages and prior work [22].

Diverse Types of Data Sources. Vantages embraces the operational practice of putting DNSKEYs on webpages as an example of a way to augment topological and protocol

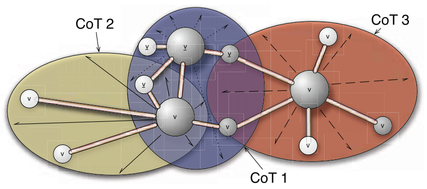


Fig. 3. CoTs will be overlapping sets of Vantages v.

diversity. Moreover, some zones like .br have put their DS records in the whois database. These practices help Vantages to detect misconfigurations in which some sources are misconfigured, but others are not. By checking keys for consistency across different protocols as well as just network Vantages, operational mistakes that are localized to one server, or one set of servers (like the name servers) can be compensated for by a larger majority of genuine values. In the case of conflicting information, clients must decide which key they will choose to use. Although diversity of vantage points gives one confidence about getting the correct DNSSEC keys when multiple ways of checking return consistent results, one must also be able to determine the correct answer in case of conflicting observations. As this paper reports a brand new design approach to key verification, we do not believe a "one size fits all" answer regarding which is the best one to use at this time; the answer may well depend on specific deployment scenarios. We propose to start simple with a scheme that is loosely based on a "majority wins" theme (called CPBUC, below), and to learn from practice whether this simple solution is adequate or why we may need a more sophisticated solution.

Incomplete and Conflicting Data. Public Data verification becomes slightly more complicated in Vantages because DNS resolvers serving different users do not always query the same DNS zones. As a result, not all witnesses necessarily try to learn the same DNSKEYs. Therefore, Vantages classifies keys according to the amount of evidence that is available from a CoT with a scheme called CPBUC. This evidence-based classification approach is an attempt to overcome missing data, conflicting data, attacks, and stale data values by expressing the tradeoffs to a user in the CPBUC policy framework, which has the following exclusive states:

- *Confirmed*. At least a threshold number (p) of witnesses have seen a key, and no conflicting values were seen from any data source or witness.
- *Provisional*. The same as the confirmed state, but where less than p witnesses have seen the key.
- *Byzantine*. When witnesses see conflicting keys for a zone, but more than a user-specified fraction of them have seen the same value ($\frac{2}{3}$ by default, named in spirit of Byzantine fault model [19]).
- *Unknown*. Keys that have not been seen by any other witnesses.
- *Conflict*. The last resort when none of the other states fit and a conflicting set of keys were seen.

Securing CoT Communications. Finally, the communications between instances of Vantages are cryptographically protected by using a daemon's PGP key to sign all observations and send the signatures with all messages to peers. The signatures also allow Vantages to meet the Public Data requirement of nonrepudiation by allowing anyone possessing an observation and its signature to always prove the observation was made by the reported Vantage daemon.

4 Security Analysis

In our model, attacks *cost* resources, which might take the form of time, money, social leverage, and so on.

4.1 Threat Model

In this paper, we focus on attacks in which the adversary has not compromised all of the name servers for a zone because in such a case the actual authority is unable to publish genuine data, and thus the authentic data are not public. Rather, we focus on a network-based attack where the authority can legitimately publish data, but the resolver and portions of its CoT may not be able to observe this genuine data.

To launch a network-based attack, an adversary (Eve) must be able to capture and replace data packets that are in transit. Thus, Eve must first be able to observe these packets from a vantage v_e that she has access to, and must then be able to interpose in communications. For example, if Eve has compromised a router in an Internet service provider (ISP), she might be able to use Deep Packet Inspection (DPI) to identify when the router is forwarding either a DNS query for a `DNSKEY` or a DNS response. Then, she would need to replace the genuine value with an invalid value. In other words, the path that communications traverse can lead to an attack if an adversary is among the Vantages that compose the path (e.g., an on-path attack).

However, the work that Eve must do to successfully subvert Alice is more complicated than intercepting a single point-to-point message. In DNSSEC, Eve must attack the *set* of servers who can each report a zone's genuine key. If she spoofs just one name server, then Alice can detect her attack by querying the other servers that serve the zone. In addition, Eve must also consider the operational feasibility for her to launch a prolonged attack. Specifically, if her attack is detected, she can expect corrective action to be taken. Thus, Eve must consider attack scenarios in which she can subvert Alice while controlling the visibility of her attack.

4.2 Attack Cost

To launch a successful attack, an adversary may have to target well-protected Vantages, for which the cost may be nonnegligible. We quantify the overall cost in terms of two component functions: 1) the cost of *acquiring* nodes $c_a(V_e)$, and 2) the *usage* cost $c_t(V_e, t)$. We note that approximating the cost of this sort of activity becomes quite difficult as one attempts to make a precise estimate. Rather than attempting to achieve this elusive goal, we simply present this formulation as a single high-level candidate cost formulation.

Acquisition. We define the *Acquisition* cost in terms of the difficulty an adversary faces in compromising a node and, thus, increasing the spread of her attack. For example, some nodes (such as core routers at large ISPs) may be difficult to access, and may take especially uncommon skill-sets and/or social engineering to compromise. Previous work [13] has noted the existence and nature of an Internet black-market economy in which (among other things) routers are rented as a commodity. Here, it suffices to say that specific routers at specific locations (such as the core of a very large transit ISP) may not be for sale, or may be sold at a premium. While the level of effort needed to obtain *specific routers* can vary widely with different targets, we begin with a simple metric as an approximation for this difficulty.

When Eve wants to attack, she intends to spoof answers between a data source S_{v_i} and a client v_j, and to do this she must control at least one vantage $v_e \in \sigma_{(i,j)}$. To succeed in an attack between the set of servers for a zone V_Z and a CoT V_{CoT}, Eve must have a set of attack Vantages V_e that can intercept response messages. However, *discovering* what nodes need to be in V_e is a component of cost too because there can be a cost in finding this out. For example, trying to identify the interfaces on an ISP's router might require social engineering, or possibly cost real money. Considering that each node in V_e may have both different acquisition and discovery costs (depending on where it is, who owns it, etc.), we propose (1) as our candidate acquisition cost function:

$$c_a(V_e) = \sum_{i=0}^{|V_e|} c_{intr}(v_i) + c_{disc}(v_i). \tag{1}$$

This expression embraces the fact that each node may potentially have a different intrusion cost $c_{intr}()$ and a different discovery cost $c_{disc}()$.

Usage. The usage portion of Eve's cost logically models the notion that Eve may have recurring costs to maintain her V_e, or perhaps faces a cost that accrues over time, and that these costs may even be nonstationary (i.e., they may vary over time). For example, if Eve is snooping traffic on a router, then that router will have to inspect its traffic (DPI). This activity will result in increased CPU load, and she might eventually be *detected* when operators investigate why a router is overloaded. Clearly, this problem is more pressing on large core routers at major ISPs than in small home offices (SOHO) routers. In this case, we make a broad generalization that Eve's cost is proportional to the rate of detection λ_{detect}. Alternately, in some cases, Eve might be paying rent for access to a router that was compromised by someone else. As above, if we make a general assumption that each element in V_e may have a different usage cost and that this cost may even vary over time, then we can model her usage cost between time $t = 0$ and time $t = n$ with

$$c_u(V_e, t) = \sum_{t=0}^{n} \sum_{i=0}^{|V_e|} \lambda_{detect}(v_i, t) + c_{rent}(v_i, t). \tag{2}$$

We can see, by inspection, that as an attack is launched for a prolonged period, or as the number of nodes needed to engage in the attack grows, these cost functions do too.

4.3 The Impact of Acquisition Cost

In order for Eve to fully subvert Alice's Community of Trust (V_{CoT}), she must be able to intercept all messages between V_{CoT} and the servers Alice is trying to reach (V_Z). From this observation, we generalize that Eve needs to be in the position to be able to partition V_{CoT} from V_Z, and she would like to minimize her acquisition costs in doing so. The lower bound on the *number* of nodes she needs to compromise is on the order of the minimum-cut set: $|V_e| = O(MinCut(v_j, V_Z))$. The intuition here is that Eve's vertices must be able to disconnect (or partition) all messages from V_Z to anyone in Alice's V_{CoT}. We have included a supplementary description of minimum-cut sets (available online). We note that the cut-set may not grow every time a source or destination is added, but sometimes it can. We therefore define $V_{cut} = MinCut(V_{CoT}, V_Z)$, and use $|V_{cut}|$ as a lower bound on the number of nodes needed for Eve's attack to succeed.

Conversely, Alice's goal is to raise Eve's cost in every way she can. More specifically, her best defense is to increase the size of her min-cut set by increasing the size and topological diversity of her V_{CoT}. In determining how to spend her resources, Eve faces a tradeoff between paying to learn which nodes need to be compromised to partition Alice from V_Z (which maximizes $c_{disc}()$), and uniformly compromising as many nodes as possible in hopes of partitioning Alice (maximizing $c_{intr}()$). Clearly, if Eve spends a lot on trying to discover which nodes to use, then she may not have enough resources to cover V_{cut}. We conjecture that given a fixed target of just Alice (and not her V_{CoT}), Eve might map the BGP AS paths between v_i and V_Z, social engineer some view of intra-AS topology of each ISP on this path, then probe and try to compromise the specific routers in each of these ISPs. It is important to note that a heavy investment in discovering nodes ($c_{disc}()$) results in a high cost, and a strong possibility of failure. This is because determining Internet path information between arbitrary source/destination pairs is nontrivial, and remains an open research area [12]. Complications include inferring inter- and intra-AS topology, path asymmetries, the divergence between BGP's control and data planes, dynamism of routing, and so on.

Eve's first thought might be to focus her efforts on the upstream ISPs of V_Z (the zone's name servers). However, this negatively impacts Eve in two ways: 1) spending the resources on these servers does not allow her to subvert traffic to other zones (which makes her attack very focused, and not extensible to multiple targets), and 2) this makes her attack much easier to detect by the zone owners. If an operator for the zone has monitoring setup for their zone (such as by SecSpider [2]), then a global attack is immediately visible and, thus, much more likely to be shutdown. Clearly, this is a feasible attack, but we consider attacks in which Eve is either focused on spoofing specific users, or at least spending her attack cost in such a way that she can be in position to attack multiple and changing sets of name servers for one or more zones.

In this work, we choose to evaluate three different classifications of adversaries: 1) General, 2) Targeted, and 3) Nation State:

General. In this model, Eve's goal is to take the set of all possible Vantages to compromise (V), and acquire as many of them as she can afford ($c_{disc}() = 0$). Of all the attack models we have considered, we propose that this one is the most relevant to general Internet users. Here, Eve has a general set of compromised nodes and may be focused on spoofing either a set of users (not just Alice), or may even be performing an unstructured attack against any target of opportunity.

We observe that Eve's chances of compromising Alice's specific V_{cut} set out of all sets of possible nodes V are the same as choosing a specific combination of $r = |V_{cut}|$ elements out of a set of size $|V| : \binom{|V|}{r}^{-1}$. The intuition for this expression comes from the following reasoning: if there exists a min-cut set of size r, Eve has r chances to guess which routers are in this set (out of all $|V|$), and if she is given no additional information, then she has an equal chance to guess this set among all other sets of size r.

We can extend this slightly to say that if Eve has $n = |V_e|$ choices (where $n > r$), then her chances are multiplied by the number of combinations that can be made with nodes outside the min-cut set $(n - r)$:

$$Probability_s(V_e) = \binom{|V|}{n}^{-1} \times \binom{|V - V_{cut}|}{n - |V_{cut}|}. \tag{3}$$

Targeted. In this type of attack, we presume that Eve has learned all of the information about the path between Alice's v_i and V_Z. Though we have mentioned that this is generally an infeasible task, we give Eve the benefit of the doubt by assuming she has accomplished this somehow. However, in this model, we assume that Eve does *not* have any path information between the rest of Alice's V_{CoT} and V_Z. This could be because members of Alice's V_{CoT} may not be common knowledge, or because it is generally unrealistic for anyone to learn this information about arbitrary Internet paths.

Nation State. Finally, we consider an adversary who chooses to spend her resources compromising the n *largest* ISPs in rank order, in the hopes of disconnecting Alice. Here, the probability of success is related to the chance that the min-cut set is included in the set V_n, the top n-most well-connected ISPs, or $V_{cut} \subseteq V_n$.

4.4 The Impact of Usage Cost

Given sufficient spread, Eve's ability to spoof messages to Alice is also limited by temporal function of opportunity. However, to illustrate the magnitude of the challenge to Eve, we focus our discussion on just the acquisition costs, and in Section 5, we show that even with this simplification, Alice still has ample protection.

5 Evaluation

In this section, we report the performance of the theoretical Public Data model in two separate evaluations: 1) through extensive simulation of the pure Public Data model and 2) through quantitative evaluation of an actual live deployment of Vantages.

5.1 Simulated Evaluation

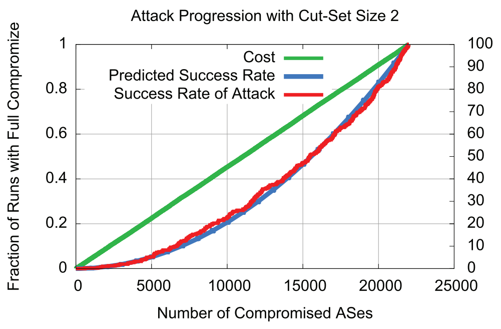
Our analysis begins by illustrating that as Alice and zone operators increase their V_{CoT} and V_Z, respectively, the cut-set V_{cut} will grow too. Next, we analyze both the probability of compromise and the progression of cost for each of our attack classes. In this work, we focus our evaluation on the acquisition cost and its probability models, and leave the evaluation of additional usage cost and its probability to future work. We claim this gives Eve a large benefit, but we will show the Public Data model still provides solid assurances.

We performed this first portion of our evaluation using a simulated Internet-like AS topology, in which each AS represents an ISP, educational institution, government agency, and so on. This topology was generated by the Inet Topology Generator [8]. Using a topology of 22,000 nodes (which is similar in scale to the current Internet), we randomly chose Vantages for Alice, her CoT, and for the servers of a target zone. We varied Alice's CoT size (including her) from 1 to 10, and for each CoT size we varied the name server set size between 1 and 10. This gave us 100 combinations. We then ran 10 simulation runs for each combination in which we varied the number of ASes that Eve had acquired from one to the full 22,000 (i.e., the whole Internet). In these attack simulations, we set the cost of compromising an AS node to be proportional to its degree (or connectivity). This is in the spirit of larger, more prominent ASes (like AT&T, Sprint, etc.) have more internal routers and more internal path diversity. Thus, for an adversary to subvert an entire AS, she will have to secure more internal routers. Moreover, larger ASes (such as tier-1 provides) are likely harder to crack into than, say, smaller ISPs. Our belief is that this linear relationship between AS degree and cost is conservative in favor of Eve.

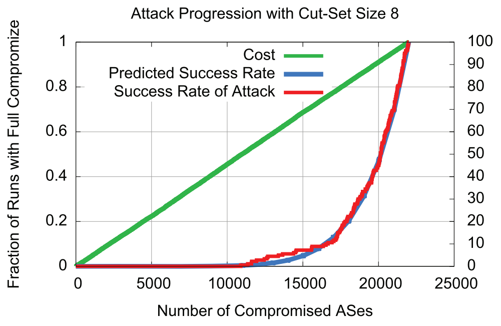
Cut-Set. Using our topology, we mapped each combination of V_{CoT} and V_Z to the min-cut set V_{cut} between them. We have included figures and a description of these results as supplementary information, available online.

Cost and Probability. Our approach in estimating Eve's costs is to be liberal in our beliefs. For example, in our evaluation we presume that for a certain "price" Eve can actually buy every AS in the Internet (which we normalized from 0 to 100). That is, we plot our simulations from one compromised AS all the way out to where Eve can arbitrarily intercept any packets into and out of every AS in the whole Internet. While we claim that this extreme is wholly unrealistic, we simply use this extreme to illustrate the scaling properties of our model in the presence of such an all-powerful adversary. We use this to gain a sense for how large Alice's min-cut sets need to be to overcome adversaries (from impoverished to very powerful) for our three classes: general, targeted, and nation state.

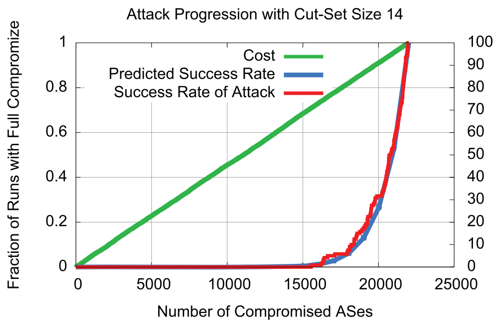
Fig. 4 shows three representative examples of the many min-cut set sizes we evaluated. Here, we can see that as a *General* adversary, Eve's success rate at subverting Alice closely follows our predictions in (3). We found that if Alice's V_{cut} is on the order of size 2, then Eve still has a sublinear chance of spoofing her. While this is certainly an important concern, we note that the growth figures (in our supplementary information, available online) illustrate that Alice can generally increases her V_{cut} size well beyond 2 (even when $V_Z = 2$) by adding witnesses to her V_{CoT}. Further, in Section 5.2, we will see that actual measurements show that many high-profile zones V_{cut} between 11 and 27, in the Internet today. In cases when V_{cut} is only size 14,



(a) Here we see a $|V_{cut}| = 2$.



(b) $|V_{cut}| = 8$.



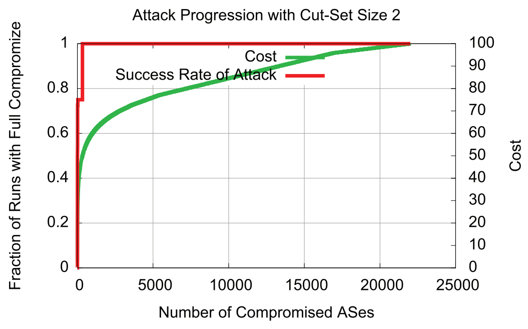
(c) $|V_{cut}| = 14$.

Fig. 4. Equation (3) very closely predicts these experimental results (for the General adversary).

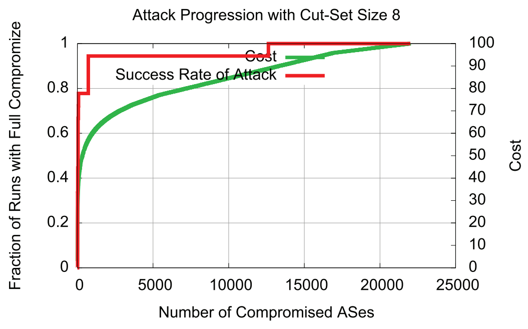
Eve's investments yield only very marginal 5 percent chance of success as her cost approaches roughly 80 percent of the cost to compromise the all ASes in the Internet.

We discuss our analysis of the *Targeted* adversary in our supplementary information, available online.

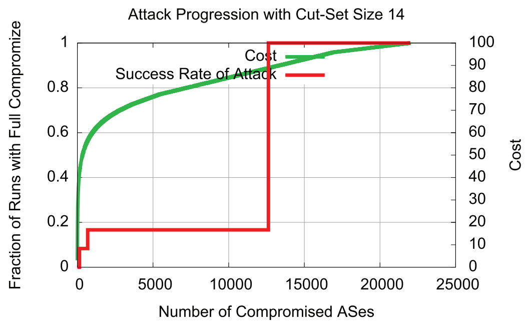
Finally, we examine the Nation State adversary, where Eve represents a well-funded organization. This is a crucial observation because we can see from Fig. 5 that the cost to her becomes on the order of that to insert herself into all traffic in and out of all ASes in the entire Internet within the first percentages of ASes that she compromises. Moreover, we can see that if Alice has a $|V_{cut}| \approx 14$, Eve spends 90 percent of the total cost of fully compromising the Internet before subverting Alice in more than 18 percent of the cases. One interesting note from this class of adversary is the odd step-function-like behavior of the graphs. This behavior comes from the fact that there are high-degree hub-nodes (which resemble tier-1 ISPs) that might intuitively seem like they *should* belong to V_{cut} in almost any situation. However, the simulations show that with a



(a) For a Rank-Order adversary, with a small $V_{cut} = 2$ Alice falls victim early in the attack.



(b) Nation State adversary with $V_{cut} = 8$, Alice is likely to fall victim to the attack early on.



(c) Nation State adversary with $V_{cut} = 14$, Eve pays a huge cost and her success is staved off, because of connectivity at the edge.

Fig. 5. Nation State adversaries against V_{cut} sizes.

sufficiently high path diversity (from a large V_{CoT}), connectivity between witnesses in V_{CoT} is pushed to peerings at the edge of the Internet. Indeed, recent measurement results [9] suggest that the edges are indeed becoming increasingly well connected. Thus, in *these* cases, a brute-force attack on the Internet's core tier-1s may still *not* subvert the protections of Public Data verification until quite a high cost is paid.

5.2 Vantages Deployment Measurements

We began running a four node CoT with Vantages witnesses in Los Angeles, Colorado, and Virginia in 2009. We also augmented this deployment CoT with the nine pollers of SecSpider. During this time, our CoT both overcame various operational hurdles in the wild that confounded DNSSEC and quantifiably outperformed its global deployment. One important note to consider is that SecSpider is not a special service. Indeed, at the time of this writing, RIPE's Atlas system [20] allows users to issue DNS queries to several thousand distinct active DNS monitoring nodes around the world, and could be used to create a Super Sized witness.

To calculate the various min-cut sets needed, we inspected the list of all zones from SecSpider's monitoring corpus. Many operational groups consider SecSpider's corpus to be the most operationally comprehensive list of DNSSEC zones available. To model the actual Internet's inter-AS topology, we used the IRL's topology [24] data. As we noted earlier, this likely only gives us an approximation of the actual connectivity, but we use it as a systematic representation of the operational Internet. To calculate routes between each source AS in V_{CoT} and the destination name server in V_Z, we used the Internet's standard no-valley routing policy.

From our analysis, we see certain intuitive differences between zones. For example, large TLDs often spend a large amount of effort provisioning, and specifically, the min-cut set sizes measured for several operational TLDs and the root zone are: $root = 27$, $.gov = 18$, $.br = 18$, $.bg = 13$, and $.org = 11$. Compared to the results used in our simulations, these are relatively large min-cut set sizes. Min-cut sets of these sizes are easily accounted for by the p values described above. To put these min-cut set sizes in perspective, we refer to Section 5.1, where we demonstrated that if Eve is a "General" type adversary, for her to spoof a V_{CoT} / V_Z pair whose min-cut set sizes is 14, her chances of succeeding are only 25 percent after spending roughly 89 percent of the estimated cost to compromise all of the ASes in the *entire Internet*. This suggests that the root zone and many of the critical TLDs are *already* well protected by Vantages. In our supplementary material (available online), we quantitatively compare deployment metrics of our Vantages deployment with the performance DNSSEC's deployment.

6 Conclusion

This work is motivated by the fact that DNSSEC is fast becoming the first operationally deployed Internet-scale distributed system that verifies itself using public key cryptography. In this paper, we argue that the DNSSEC design must recognize the distinction between the lookup and verification processes, be robust against data delivery failures, continue to function amidst imperfect operations, and tolerate gaps in its key learning hierarchy. This becomes even more essential if groups such as IETF DANE [7] working group make more extensive use of the global system that DNSSEC provides.

To address these challenges, we propose to augment DNSSEC validation by introducing both a theoretical model called *Public Data* and open-source implementation of it called *Vantages*. *Public Data* are a novel evidence-based trust model that makes three main contributions. First, it formally defines a new (noncrypto) substrate for verifying cryptographic keys. Second, by using measurements taken among trusted data sources, it allows zone operators to augment their own data redundancy in such a way that resolvers are now able to continue to verify keys even in the presence of zone misconfigurations. And finally, using (3), we provide operators the ability to calculate their exposure.

From Fig. 4, we can see that the distributed consensus approach offers quantifiable assurances against adversaries and that the simulated results closely match the predicted model. Under reasonable DNSSEC and CoT assumptions, an adversary may need to be able to perform on-path attacks to all data traffic (into and out of) up to 90 percent of all ASes in the whole Internet before having even a 10 percent chance of spoofing a DNSKEY.

We also used our model to make qualitative and quantitative contributions to DNSSEC by implementing and deploying *vantages*, as it addresses the operational challenges ranging from using diverse data sources (DNS, web, etc.) to aligning costs with incentives for deployment. It has been able to properly verify its data more than twice as well as DNSSEC (0.964 versus 0.437).

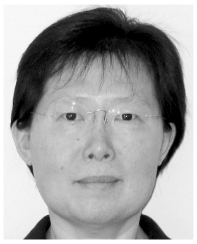
REFERENCES

[1] "North American Network Operators Group (NANOG),"http://www.nanog.org/, 2013.
[2] SecSpider, http://secspider.cs.ucla.edu/, 2013.
[3] Vantages, http://www.vantage-points.org/, 2013.
[4] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS Security Introduction and Requirement," RFC 4033, Mar. 2005.
[5] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "Protocol Modifications for the DNS Security Extensions," RFC 4035, Mar. 2005.
[6] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "Resource Records for the DNS Security Extensions," RFC 4034, Mar. 2005.
[7] IETF. DANE), https://datatracker.ietf.org/wg/dane/charter/, 2013.
[8] C. Jin, Q. Chen, and S. Jamin, "Inet Topology Generator," Technical Report CSE-TR-456-02, Univ. of Michigan, 2000.
[9] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet Inter-Domain Traffic," *Proc. ACM SIGCOMM '10,* 2010.
[10] M. Larson, "[dnssec-deployment] rrsig for arpa expired," *DNSSEC-Deployment Mail List,* http://dnssec-deployment.org/pipermail/dnssec-deployment/2010-June/004009.html, June 2010.
[11] V. Levigneron, "Key Deletion Issues and Other DNSSEC Stories," *Proc. Int'l Conf. Artificial Neural Networks (ICANN '41),* June 2011.
[12] Z.M. Mao, J. Rexford, J. Wang, and R.H. Katz, "Towards an Accurate As-Level Traceroute Tool," *Proc. ACM SIGCOMM '03,* 2003.
[13] J. Martin and R. Thomas, "The Underground Economy: Priceless," *USENIX; login,* vol. 31, no. 6, pp. 7-16, 2006.
[14] P. Mockapetris and K.J. Dunlap, "Development of the Domain Name System," *Proc. ACM SIGCOMM '88,* 1988.
[15] E. Osterweil, "Measurable Security: A New Substrate for DNSSEC: Chapter 3.1.1 (Trials of the .gov TLD)," PhD dissertation, UCLA, Dept. of Computer Science, http://irl.cs.ucla.edu/eoster/doc/osterweil_thesis.pdf, 2010.
[16] E. Osterweil, D. Massey, B. Tsendjav, B. Zhang, and L. Zhang, "Security through Publicity," *Proc. First USENIX Workshop Hot Topics in Security,* 2006.
[17] E. Osterweil, M. Ryan, D. Massey, and L. Zhang, "Quantifying the Operational Status of the DNSSEC Deployment," *Proc. Eighth ACM SIGCOMM Conf. Internet Measurement (IMC '08),* 2008.
[18] G. Patrick, "Re: [dns-operations] Expired DNSSEC Signatures in .gov," *dns-operations,* http://www.mail-archive.com/dns-operations@lists.dns-oarc.net/msg00661.html, Sept. 2012.
[19] M. Pease, R. Shostak, and L. Lamport, "Reaching Agreement in the Presence of Faults," *J. ACM,* vol. 27, no. 2, pp. 228-234, Apr. 1980.
[20] RIPE. Atlas). http://atlas.ripe.net/, 2013.
[21] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," *Proc. ACM SIGCOMM '01,* 2001.
[22] D. Wendlandt, D. Andersen, and A. Perrig, "Perspectives: Improving SSH-Style Host Authentication with Multi-Path Probing," *Proc. USENIX Ann. Technical Conf.,* 2008.
[23] Wikiquote, "Yogi berra — wikiquote,"http://en.wikiquote.org/w/index.php?title=Yogi_Berra&oldid=607127, 2007.
[24] B. Zhang and R. Liu, "Collecting the Internet As-Level Topology," *ACM SIGCOMM Computer Comm. Rev. (CCR),* vol. 35, pp. 53-61, 2005.
[25] P.R. Zimmermann, *The Official PGP User's Guide.* MIT Press, 1995.

Eric Osterweil received the PhD degree from the Computer Science Department of UCLA in 2010, where his dissertation topic was a new substrate for Internet-scale security systems called "Measurable Security." He is an applied security researcher in the CSO office at Verisign. His research interests include measuring the deployments, systemic dependencies, and security of Internet-scale systems like DNS, DNSSEC, and BGP.

Dan Massey received the doctorate degree from UCLA. He is an associate professor in the Computer Science Department at Colorado State University. He is currently the principal investigator on research projects investigating techniques for improving the Internet's naming and routing infrastructures. He is a coeditor of the DNSSEC standard (RFC 4033, 4034, and 4035). His research interests include protocol design and security for large scale network infrastructures. He is a senior member of the IEEE, IEEE Communications Society, and IEEE Computer Society.

Danny McPherson is the chief security officer (CSO) at VeriSign, Inc. He is with the ICANN Security and Stability Advisory Council (SSAC), the ICANN DNS Root Server System Advisory Committee (RSSAC), the FCC's Network Reliability Interoperability Council (NRIC), and is a voting member of the FCC's Communications Security, Reliability and Interoperability Council's (CSRIC). He is a member of the Internet Architecture Board (IAB), Internet Research Steering Group (IRSG), and cochairs the IETF's L3VPN WG.



Lixia Zhang received the PhD degree in computer science from MIT and joined Xerox Palo Alto Research Center as a member of research staff in 1989. She is currently a professor in the Computer Science Department at UCLA. In the past, she served as the vice chair of ACM SIGCOMM, member of the editorial board for the *IEEE/ACM Transactions on Networking*, member of the Internet Architecture Board, and co-chair of the Routing Research Group under IRTF. She received the IEEE Internet Award in 2009 and holds the UCLA Jon Postel chair in computer science. She is a fellow of the ACM and IEEE.