Research Statement

Dave Levin

My research interests are in the areas of systems and network security, with an emphasis on achieving cooperation in the presence of self-interested parties. Originally, the Internet was a technological playground, a collaborative endeavor among researchers who shared the common goal of achieving communication. Malice and self-interest used not to be concerns, but today, the Internet consists of millions of commercial entities and over three billion users who often have conflicting goals. For example, some countries actively censor or monitor not only their own citizens' traffic, but *all* traffic that merely transits their borders, raising concerns internationally of the integrity and confidentiality of user data—all because the Internet happened to route traffic through a country with a competing interest. As another example, some certificate authorities issue certificates for free but, in an effort to offset bandwidth costs, charge their customers to revoke, introducing a perverse economic incentive for website administrators *not* to revoke compromised certificates, and leaving users at risk of impersonation attacks—all to reduce one party's bandwidth costs. These examples demonstrate that protocols that are merely technologically proficient are not enough. Successful networked systems must account for the misuse and abuse that arise from potentially competing interests.

My work investigates the practical application of **economic theory** and **cryptography** to study and build networked systems that treat **users' incentives and abilities** as first-order design principles. As an interdisciplinary researcher, I apply techniques from these areas to designs for which I can rigorously *prove* that participants can safely and efficiently coexist, even with competing interests. Experience dictates that relying upon heuristics alone opens the door to selfish or malevolent acts; conversely, my approach is to analyze systems with empirical measurements and game theoretic or cryptographic models to rigorously demonstrate *why* participants act a particular way.

The complexities of a networked system cannot be captured by theory alone. I believe the best way to prove a system will work in practice is to **build it**, and the best way to understand an existing system is to empirically **measure it**. My students and I have built systems spanning areas such as large-scale peer-to-peer networks, wireless networks, and trusted hardware, and have analyzed large-scale systems such as Bitcoin, the Web's certificate ecosystem, a root DNS server, and the Tor anonymity network. In so doing, I have discovered that empirically measuring network security can expose problems that have immediate impact on users, and that applying theoretical techniques to practical systems refines the theory and engenders new system mechanisms.

Below, I discuss what I believe to be some of my strongest contributions in building systems that account for competing interests. My contributions in this area can be naturally grouped into two broad agendas: measuring network security and building new primitives for secure cooperation. I have found that these two approaches complement one another well; I apply measurement results to inform my systems' designs, and I develop systems that can permit greater understanding and control over the network.

What can we learn by empirically measuring network security?

In my experience, developing systems that manage competing interests greatly benefits from empirically understanding the motivations and capabilities of those who ultimately run those systems. To this end, my systems-building is driven by empirical measurements. I have performed measurement studies on the root DNS server hosted at UMD [2], the Tor network [1], the Bitcoin topology, and in an ongoing effort, the Web's public key infrastructure (PKI). Here, I discuss two of my measurement efforts—inferring how well the PKI supports certificate revocations and measuring latencies on the Tor network—and how they form the concrete basis for my future work.

How is the Web's PKI administered? The importance of the Web's PKI cannot be overstated: its certificates provide users with the ability to verify with whom they are communicating online, and it enables encryption of those communications. While the online *use* of the PKI is mostly automated, there is a surprising amount of human intervention in *management* tasks that are crucial to its proper operation. Obtaining, reissuing, and revoking certificates are often cumbersome and expensive processes, and downloading revocation information can increase bandwidth costs and page load-times. Yet, to ensure the Web's security, users need administrators, certificate authorities, and browsers to all do their part.

Along with colleagues at Northeastern University and Duke/Akamai Technologies, I have been working towards understanding and improving how the Web's certificates are administered. We have performed the first end-to-end studies of certificate revocation, asking: do administrators revoke compromised certificates [6], and do browsers bother obtain-

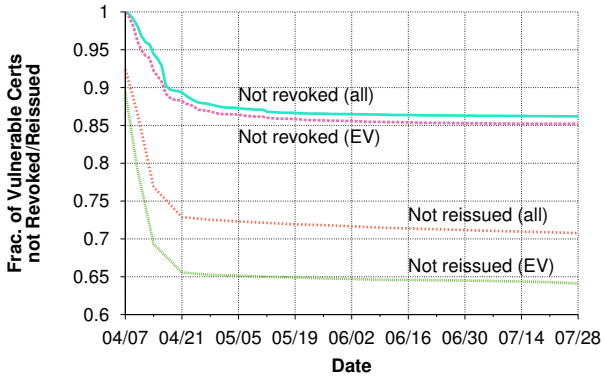

Figure 1: More than three months after Heartbleed (April 7, 2014), most of the top-1M most popular websites failed to revoke and reissue their vulnerable certificates [6]. (Note that the y-axis does not start at zero.) This motivates future work to improve the Web's PKI.

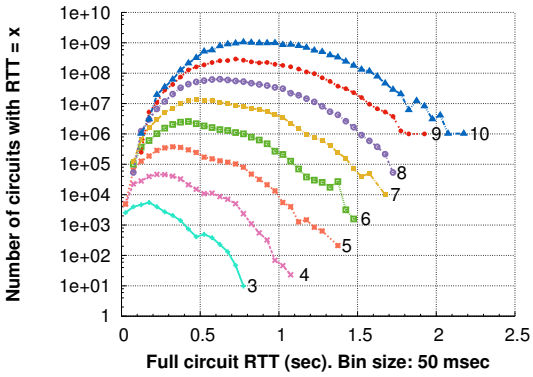

Figure 2: Ting [1] shows that longer Tor circuits have more options for lower- and higher-latency paths. (Each line is annotated with its corresponding circuit lengths.) This encourages future work on more sophisticated circuit selection.

ing them? Measuring administration at a global scale is an extremely challenging problem; a survey would be difficult to perform, and some administrators may not have incentive to answer truthfully. Instead, we used the widespread Heartbleed vulnerability as a sort of *natural experiment*: it gave us a specific day and time at which all vulnerable certificates *should have* been revoked and reissued. While almost all vulnerable servers patched their software (93%), Figure 1 shows that certificate management lagged far behind: in this plot, the x-axis begins on the day that Heartbleed was announced, and the y-axis shows how many vulnerable certificates had *not yet* been revoked or reissued by day x. Ideally, this plot would be a vertical line on the y-axis, indicating complete and immediate adherence to correct security policies by all website administrators. Instead, it shows a sharp but short-lived reaction; after more than three months, only 29% of vulnerable certificates were reissued, and a measly 14% were revoked. Moreover, we found that *no browsers* fully check for all certificate revocations, and mobile browsers in particular do no checking whatsoever. Our code and data are available at https://securepki.org

I believe strongly in sharing results with wide ranges of audiences, and I have presented these findings to administrators, network operators, CISOs, and developers of international standards for maritime communication. Through these interactions, I have come to find that this lapse in certificate management is driven in large part by misaligned economic incentives: some certificate authorities charge their customers to revoke (to recoup bandwidth costs), and browsers eschew revocation checks in an effort to minimize page load times. One of my goals in my future work is to secure the Web's PKI. Doing so will require explicitly accounting for these economic (de)motivators.

What do latencies reveal about anonymity networks? As researchers, I believe we must not only develop new systems and mechanisms, but we must also understand how the successful, deployed systems operate and can be improved upon. I advised an undergraduate student in the development, validation, and application of **Ting**, a system that can accurately measure the latency between *any* two nodes on the Tor anonymity network. We showed that Ting is highly accurate (91% of the time, its error is less than 10%), and that its measurements are consistent over time, permitting us to collect an all-pairs latency dataset among a 50-node subset of the live Tor network.

Beyond being a generally useful network measurement tool, the results from Ting form the foundation of several interesting areas of potential attacks on and improvements to the Tor network. For instance, we showed how Tor latencies can be applied to speed up deanonymization attacks by a factor of $1.5\times$. We also found that it is possible to use circuits longer than Tor's default of three hops without necessarily imposing very high latencies. Figure 2 shows that there are orders of magnitude more 10-hop circuits with less than 200 msec latency than there are such 3-hop circuits. While there are still many open questions—for instance, do longer circuits increase the chances of attackers being on the path, and does this help attackers deanonymize?—these empirical measurements show that there need not be a strict trade-off between circuit length and latency, thereby motivating future work towards more sophisticated circuit selection.

Can we make (some) attacks impossible?

Designing and building secure, scalable systems is extremely difficult, often requiring a careful choice of trade-offs among performance, cost, and the types of attacks one can defend against. What makes this process all the more challenging is that it can be difficult to anticipate the motivations and capabilities future attackers may have: Tor, for example, was originally designed for anonymous communication, but has had to evolve to provide availability in the presence of censoring regimes. My work seeks to avoid the cat-and-mouse game of security by developing new building blocks that provide proof that an attacker *could not* have performed certain actions. Armed, for instance, with the knowledge that a given attacker could not have seen let alone manipulated a user's in-flight packets, designers can rule out many potential attack vectors and thus more easily reason about their systems' security.

The first malicious act I made impossible was "equivocation." Equivocation is a very simple, seemingly innocuous act: it simply means sending conflicting messages to others. Though simple, it can be powerful, and it is the basis of many attacks. Notably, the Byzantine generals problem can be solved with only a simple (as opposed to two-thirds) majority if no participants were able to equivocate. Along with my colleagues at Microsoft Research, I designed and implemented **TrInc** [3], the smallest piece of trusted hardware that can solve equivocation. Of course, I mean functionally small: TrInc consists only of a monotonically non-decreasing counter (thus the name: Trusted Incrementer), and a cryptographic key for generating attestations. The main insight behind TrInc is that, as long as the counter cannot decrease, the piece of hardware (which we call a *trinket*) generates unique attestations. By giving semantic meaning to the counter (e.g., letting it represent the number of blocks downloaded so far in a BitTorrent swarm), the unique attestations render equivocation impossible. We applied TrInc to over a dozen systems, and I implemented three of them: a trusted, append-only log, a simplified version of an accountability system (PeerReview), and a solution to an open incentives problem I had identified in BitTorrent [4].

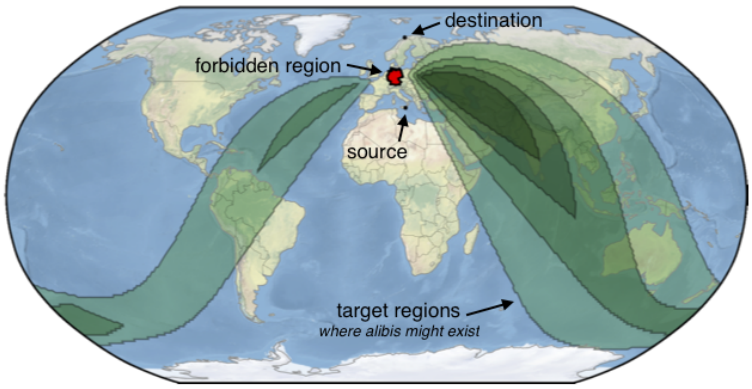

Figure 3: Alibi Routing [5] allows users to request that their packets provably avoid geographic regions of their choosing. Here, a host in Italy wishes to communicate with a host in Norway while avoiding Germany. Alibi Routing calculates targets regions and searches within them for potential alibi peers.

Another "impossible" primitive I have designed addresses the problem of online censorship. Censorship can take many forms, such as explicit blocking of traffic, injecting packets with false information, or logging users' data to be used against them. As such events have become more common and more public, user demand for control over what happens to their traffic has grown. Working with several graduate and undergraduate students, I developed **Alibi Routing** [5], an overlay system that allows users to specify regions of the world they wish their packets to avoid, and returns *proof* that it was able to avoid those forbidden regions. In general, proving that something did not happen can be extremely difficult, sometimes requiring one to enumerate and rule out everything that could have happened, but such an approach would be infeasible, as there does not exist a single, trustworthy map of the Internet. Alibi Routing introduces a new proof structure: it proves (with a symmetric key MAC) that a packet visited a peer at a known location, and it verifies (with latency measurements) that the time it would have taken to go through that peer *and* any point the forbidden region would have required information to travel faster than the speed of light, which would have been impossible. The peer therefore acts as an *alibi*. Alibi Routing shows that this proof structure can be applied in a scalable system, and I believe it can be applied to improve Tor and network provenance systems. I am a strong advocate of making code and data publicly available; these are available for Alibi Routing at https://alibi.cs.umd.edu

Future directions

One of the most impressive features of the Internet is that we can achieve global connectivity even while service providers maintain autonomous control over their own networks. However, this decentralization comes at a cost: User choice is vastly limited, subjecting many users to service providers' policies, which may be misaligned with their own goals or interests. My work is motivated by the principle that it is not only possible to enable users with greater control over their data, but that doing so can result in a more secure, more efficient Internet for all (non-malicious) parties. My future research agenda will seek to demonstrate this in two broad ways. First, I will perform wide-scale measurements to develop a more complete view of how the Internet's security is *truly* (mis)managed, with an initial emphasis on what I observe to be an increased centralization of the Web's PKI. Second, I will design systems that give users greater insight into and control over how their data and identities are disseminated throughout the network.

Securing the Web's PKI. The semantics of authentication in the Web's PKI are rather straightforward: if there is a certificate binding Alice's name to a public key, then (barring key compromise) anyone who can prove knowledge of the corresponding private key must be Alice. However, in reality, the majority of websites—especially many of the most popular ones—are hosted at least in part by third-party Content Distribution Networks (CDNs, e.g., Akamai) or web hosting services. Put simply: administrators of websites who deal with critically sensitive user data are *giving their private keys away* to third parties.

This has the potential to result in widespread attacks that could compromise the security of virtually all Internet users' online banking, e-commerce, and social networking data. Sharing keys with the large, popular CDNs and hosting services places an *immense* amount of power in the hands of a few hosting providers—they (or any attacker who could infiltrate their networks) could arbitrarily impersonate any of their customers, even over HTTPS. Because of the economic incentives of third-party hosting, such key sharing is endemic, and yet it is surprisingly not widely known by the public, and has received very little attention from the research or network security communities.

I am working towards performing the first wide-scale measurement study of key sharing with third-party hosting services. One of the reasons that websites share their private keys—and thus their users' data—with third parties is that there are no available alternatives that balance the demands of both the users and the service providers. I will also investigate new protocols that seek to maintain the benefits of a third party without having to divulge users' private data. The challenge with such a protocol is that CDNs need to view *some* of the data in plaintext in order to perform some in-line security checks (e.g., for SQL injection attacks). I envision this work benefiting from my ongoing and future collaborations with network measurement researchers, CDNs, and cryptographers.

New cloud primitives for persistent mobile security. Today's cloud computing offers two broad types of computation: arbitrary VMs with a large footprint, and constrained software-as-a-service APIs with a small footprint. Yet neither of these types of service meet the needs of what I believe to be a growing need for services that are personalized, have low overhead, and yet demand persistent run-time, such as a personalized email server or a security-enhancing proxy. Users' security could benefit greatly from such proxies—for instance, by performing certificate revocation checking when the user is mobile—but it would be economically prohibitive for most users to run a VM in a compute cloud all-day, every-day. In other words, user security suffers because of a lack of incentives for a cloud provider to deploy a service that would permit users to affordably run always-on, personalized services.

Along with colleagues from Northeastern, Duke, and students at UMD, I have been developing a new process-based primitive for cloud computing: users submit arbitrary processes, and our system provides the abstraction that the process is always running, while in reality it can swap the process to cold, long-term storage. We maintain our always-running abstraction by preemptively swapping in only the pages that the process absolutely needs to run. I believe such an abstraction will make personalized services more usable (due to its persistence) and more economically viable (due to its low resource consumption). I plan to use this foundation to develop systems that take more complicated security operations (e.g., certificate revocation checking and server-side obfuscation of data access) and offload them to a small cloud-based process that migrates to always be close to a mobile user.

A new routing primitive for the Internet. Internet users throughout the world have expressed varying concerns over how their data is treated while in transit: those near a censoring regime are sometimes subject to the same censorship as if they were within that regime's borders, and packets that transit some countries' borders are subject to bulk data collection. I envision building and deploying systems that enable users with the insight into and control over where their data does and *does not* go. Our Alibi Routing system took the first step towards this by providing packet-by-packet proofs of avoidance, but it is not yet clear how this translates to the higher layers of the network stack: for instance, if one transmission of a packet does not yield a proof of avoidance, how should TCP and applications (e.g.,

HTTP) react? Is it possible to extend this control to ensure that a user can keep a given entity from *ever* viewing his or her data? One possible approach would be to combine multiple paths that mutually avoid one another, and slice information across them (e.g., using Shamir's secret sharing). My goal is a widespread deployment, which could have far-reaching impact on policy and network administration by shedding light on a basic but difficult question to answer: who wants to avoid whom?

References

[1] F. Cangialosi, D. Levin, and N. Spring. Ting: Measuring and Exploiting Latencies Between All Tor Nodes. In *ACM Internet Measurement Conference (IMC)*, 2015.

[2] M. Lentz, D. Levin, J. Castonguay, N. Spring, and B. Bhattacharjee. D-mystifying the D-root Address Change. In *ACM Internet Measurement Conference (IMC)*, 2013.

[3] D. Levin, J. R. Douceur, J. R. Lorch, and T. Moscibroda. TrInc: Small trusted hardware for large distributed systems. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.

[4] D. Levin, K. LaCurts, N. Spring, and B. Bhattacharjee. BitTorrent is an auction: Analyzing and improving BitTorrent's incentives. In *ACM SIGCOMM*, 2008.

[5] D. Levin, Y. Lee, L. Valenta, Z. Li, V. Lai, C. Lumezanu, N. Spring, and B. Bhattacharjee. Alibi Routing. In *ACM SIGCOMM*, 2015.

[6] L. Zhang, D. Choffnes, T. Dumitraş, D. Levin, A. Mislove, A. Schulman, and C. Wilson. Analysis of SSL Certificate Reissues and Revocations in the Wake of Heartbleed. In *ACM Internet Measurement Conference (IMC)*, 2014.