Thesis Proposal — DRAFT

Zachary Weinberg

September 3, 2013

Abstract

Traffic analysis is specifically excluded from the standard threat model for a secure channel. We claim that this has become unrealistic, and that failure to include traffic analysis in the threat model leads to exploitable vulnerabilities "in the wild." We propose to study these vulnerabilities and develop countermeasures.

1 Background

Traffic analysis is the craft of extracting information from a secure channel, using whatever data is still observable despite encryption. Often this is metadata, such as the source, destination, establishment time, and duration of the connection, and the protocol in use. In a typical modern packet-switched network, it is also possible to observe when each individual packet is sent and the size of its encrypted payload, and to perform TCP stream reassembly. The adversary may also be able to correlate observations of network traffic at more than one point in the topology, or to correlate network events with other events. For instance, if the adversary observes a large client-server upload to commons.wikimedia.org at the same time as a new image appears on that site, they may deduce that the image was uploaded by the client under observation.

Traffic analysis has been practiced since the Great Game of the 19th century, if not longer, but came into its own with the use of radio for military communications in World War I. French intelligence was able to locate German encampments via radio direction finding, and identify transmissions as originating from HQ, infantry, or cavalry divisions because each produced a different volume and pattern of traffic. In the Pacific theater of World War II, both sides were able to make educated guesses at upcoming naval maneuvers from radio traffic volume, changes in callsigns, time correlation of signals, and so on. [26]

Nowadays, traffic analysis is more often applied to messages on a network of some sort. Felten [20] observes that the mere fact of a call placed to certain special-purpose telephone numbers reveals "basic and often sensitive" information about the caller. There are dedicated support hotlines for survivors of domestic violence, people considering suicide, and sufferers from various forms of addiction; there are also lines dedicated to anonymously reporting government misconduct, tax fraud, illegal firearms, and other crimes. If phone call records are analyzed over time, they reveal the "social graph" and can even indicate the nature of personal connections; Felten observes that regular late-night phone calls suggest a long-distance intimate relationship. The "dragnet" surveillance programs (PRISM, X-KEYSCORE, etc.) recently revealed to be conducted by the United States' intelligence apparatus [22] rely on traffic analysis as a first-stage filter: phone numbers registered outside the USA, calls to people known to have called people on various "watch lists," apparent use of encryption, etc. all trigger more aggressive data collection.

1.1 Unlinkability

Because it is so easy to extract information from the source and destination of encrypted communications, protocols have been developed specifically to conceal this information. In addition to all the properties of a secure channel, an *unlinkable* channel protocol seeks to ensure that no eavesdropper at a single point in the network can learn *both endpoints* of the channel. That is, no eavesdropper can tell that Alice is talking to Bob over an unlinkable channel, but eavesdroppers can still tell that Alice and Bob are talking to *someone*. (This definition is due to Pfitzmann [31]. Whether or not active attackers are considered depends on the writer.)

The standard technique for providing unlinkability, invented by Chaum [9], is to pass each message through one or more "relays," layering encryption so that each relay only knows the preceding and next link in the chain. One relay is enough against most eavesdroppers, but provides no protection if the relay itself is malicious; operators of popular single-hop "proxy services" or "anonymous remailers" may, and indeed have, come under coercion to expose their users. [2, 30, 37] Similarly, a two-hop chain is vulnerable to a pair of cooperating malicious relays. In principle, a three-hop chain is secure unless the adversary controls a significant fraction of the network, and more hops only add overhead [39, 40].

Chaum's original design was geared for email, so it imposed significant delays at each relay in order to conceal the exact time of each message. This delay is unacceptable for interactive applications (such as online chat and Web browsing), and the current generation of "low latency" unlinkable services, such as Tor [18], deliberately omit it. This opens an avenue for "intersection" attacks by adversaries who *do* control a significant fraction of the network, or who can observe traffic at many points in the network; these are

formally out of scope, but have still received a good deal of attention in the literature, e.g. [13, 14, 15, 16, 29, 36, 44] Defenses against these attacks are still being researched, although a few simple tactics have been deployed, such as "guard nodes" [44]: the attacks are most effective if a malicious node is directly connected to a client, so by always beginning one's chains with the same node for an extended period, one avoids having any of one's traffic be exposed to a malicious node (assuming that the initial choice is sound).

1.2 Confidentiality of Message Length

The standard mathematical definition of "confidentiality" assumes that all messages are the same length. In practice, this is not true, and as recently demonstrated [21, 34], that permits serious active attacks on protocols that compress data before encryption. But even a pure eavesdropper can learn something from the size of a message.

HTTP (over any secure channel, unlinkable or not) is particularly vulnerable to this attack because it exhibits strict turn-taking behavior: a single TCP connection can be reused for several query-response pairs, but the client cannot begin a new query until it has received the complete response to its previous query, nor can the server begin a response before the query is fully received.[1] Thus, the approximate length of each request and its corresponding response is apparent to an eavesdropper capable of TCP stream reconstruction. Furthermore, most web pages incorporate by reference a set of "resources," such as images, style sheets, and scripts, which must be loaded in separate HTTP transactions (unless cached); thus two pages of approximately the same length may be distinguishable because they refer to different resource sets.

We will give three examples of concrete breaches of confidentiality via message length analysis. All these attacks assume an adversary eavesdropping directly, or perhaps at one or two hops' remove, upon a specific, known target user who is browsing the Web. The simplest attack observes that the vast majority of Facebook and Google profile images have a unique size. Detecting the HTTPS request for this image, and measuring its length, thus permits an eavesdropper to associate a particular social-network identity with a particular IP address. [23, 33] A more sophisticated "state tracing" attack seeks to reconstruct a sequence of page loads from the lengths of HTTPS traffic bursts. Since the structure of a public website is known to the attacker, this reveals how the targeted user is interacting with the site. Applied to a search engine, this can reveal queries [7]; applied to a tax-preparation website, this can reveal the target's approximate income, marital status, and other such highly confidential data. [43] Finally, if the targeted user is browsing the Web via an unlinkable channel, an eavesdropper may be able to identify the *site* that is being accessed from the pattern of response sizes. This possibility has received substantial attention in the literature, e.g. [6, 11, 12, 19] but it is unclear to us whether it can be scaled from these laboratory experiments (typically covering at most a few thousand sites' front pages) to the entire Web.

The basic defense against an eavesdropper extracting information from message length is, of course, to pad messages. However, if done poorly, padding won't help at all; for instance, padding each packet up to the nearest multiple of 512 or 1024 bytes adds only a trivial amount of uncertainty to the length of a much longer message. If done more thoughtfully, padding can help, but often at unacceptable cost in transmission time. The best known techniques rely on application-layer knowledge of the full dataset whose members are to be indistinguishable (or clustered into indistinguishable groups). [4, 10, 28] While this may be practical for e.g. Facebook to apply to all its profile photos, it is no "magic bullet" that will fix all the sites on the net. (Compare the notion of ideal steganography [24]; it works great provided you know the true distribution of covertexts, which in practice can't even be modeled well.)

1.3 Unobservability

An *unobservable* protocol would have the property that no eavesdropper could tell that anyone was talking at all. This is a very strong property, approached by some physical radio encodings [32, 42] but not actually achieved when there is any cleartext structure to the data transmitted [25], or if all transmitters don't continuously broadcast at maximum gain (which is obviously undesirable). In packet networks, where it is not *possible* for all nodes to transmit continuously all the time, unobservability is generally not even considered. A weaker definition is for no eavesdropper to be able to tell whether any node is the *originator* of a message. Even this weaker notion, at least to date, requires too much "dummy" traffic to be considered practical, and the topic has been neglected.

Unobservability, in the strong sense, is the only theoretically-sound defense against an arbitrarily powerful traffic-analytic attacker, so we suggest that the topic should be revisited, with specific attention to whether the weak definition will suffice against realistic attackers, and how much of that "dummy" traffic can be made to be useful.

2 Research Questions

We propose to investigate the extent to which traffic analysis really does destroy confidentiality, and how practical it is to do something about it. Concretely, we plan to investigate the following three questions.

[1] HTTP 1.1 includes a *pipelining* mechanism that lifts this restriction, but unlike the rest of protocol 1.1, pipelining has never seen wide adoption. In the past few years, radical revisions of HTTP which also lift this restriction, such as Google's SPDY, have been proposed and adopted to some extent; we are not aware of any research into how this changes the situation for an eavesdropper.

To what extent does size-and-timing analysis destroy confidentiality? As discussed above, this has been studied to some extent, but mostly in "laboratory" settings. The most concrete results are all to do with single sites: learning search queries, user profile images, answers to questions on a question-and-answer site (e.g. for tax preparation). Except for profile images, these involve active interaction with the site. One avenue for further exploration would be to study whether "passive" browsing on a large, complicated site reveals interesting things about the user. Wikipedia has lots of innocuous articles, a fair number that one might be embarrassed to admit one has read, and a handful that are so politically or culturally sensitive that they cannot even be referred to in some circles. If these pages (or clusters of pages) can be distinguished by content length or detailed packet timings, that enables the adversary to learn something interesting. Similarly, social-blogging sites such as Tumblr carry an enormous range of content, with op-ed columns, art, kittens, personal journals, and pornography all jumbled up together. Facebook is complicated enough that it is likely to present a radically different network traffic profile for different logged-in users. (No study to date has looked at logged-in users in detail.)

When an unlinkable channel is used, the adversary needs to identify the site being accessed before they can do any of the above, and it may be enough for them to know that a particular site is being accessed. (Existing "Internet filtering" programs, however motivated, rely on blocking of entire websites as their principal tactic. [1]) While this too has been studied, it has not been studied in what we would consider a realistic *field* setting. Specifically, most of the literature attempts to discriminate the front pages of a few hundred servers, and examines only one traffic source, whereas country-scale "filtering" applies to millions of clients and and must consider *all* servers worldwide as potential sources of undesirable material. The number of servers that actually do carry undesirable material is much smaller, and only some of those come to the notice of the censor, but the list still potentially extends to tens of thousands of addresses. Furthermore, front pages, which express the site's corporate identity, are plausibly more different from each other than internal pages, which are often a stock set of "chrome" wrapped around a blob of text.

As a first step toward a more realistic field study of site identification, we propose to assess a much larger sample of sites, drawing on sources beyond the Alexa top *N*'s front pages, such as:

- URL shorteners contain links to material that people thought worth publicizing, indiscriminately as to topic. (It will probably be necessary to manually categorize the material.)
- "Scraping" sites operated for the specific purpose of sharing links of interest, often with categorization, such as Digg, Reddit, Metafilter, and (some areas of) 4chan.
- Special-purpose directories of sites, such as the Hidden Wiki, which lists sites operated as Tor hidden services (hidden services conceal the identity of the site operator, so these sites are more likely than the average to carry controversial content).
- Programmatically traversing internal and external links on sites revealed by all of the above.
- Exhaustive "crawls" of sites of particular interest, e.g. Wikipedia.

How practical is it to conceal within-site details via application layer countermeasures? While there are concrete proposals for semi-automated detection and masking of "side channels" (including resource length) in Web applications [4, 8, 43], to our knowledge no one has attempted to deploy them on a large public website. We are considering a detailed case study of such a deployment, probably on Wikipedia (whose developers have indicated interest in aggressive anti-surveillance measures).

It would first be necessary to produce a threat model, enumerating all potentially-sensitive pieces of information that might be revealed via traffic analysis (not just via message length), such as:

- Which page is being visited?
- Which *category* of page (main, Talk:, User:, etc; topic clusters) is being visited?
- What language? (Currently exposed via server hostname, but might not be forever.)
- Is this IP address making edits?
- Is this IP address accessing WP as a logged-in user? Do they have administrative privileges?

We would then explore ways to conceal this information. Some things can be done immediately, such as deployment of SPDY, which (at least theoretically) eliminates the strict turn-taking pattern of HTTP and therefore should make the pattern of subresource loads for each page less obvious; with some tuning, it might be possible to eliminate that pattern entirely. Other simple changes to server configuration should also help, such as making sure that the HTTP request and response headers do not change size upon login. The main project, though, would be (with the assistance of the Wikimedia organization) to implement, test, and deploy a scheme for automated dataset-aware padding of encyclopedia pages. This would not necessarily make *all* of the pages indistinguishable, but it would ensure that there were enough possibilities for each page load that learning anything interesting from each event was infeasible. It could also make sense to precache resources that might or might not be needed, such as scripts and style sheets for the page editor, in order to disguise whether the editor was ever used. (Note however that it may be infeasible to hide the fact

that an IP address has just made an edit, since this intrinsically involves transmitting the new text to the server, so the HTTP request cannot help but be larger than normal.)

Can we bake traffic-analysis resistance into the next generation of Internet protocols? It is probably not feasible to eliminate traffic analytic attacks on the present generation of Internet protocols. There are too many ways in which critical state is exposed to the network, and too many basic mechanisms can barely handle the amount of security that's already been bolted onto them. The next-generation protocols currently being designed (e.g. XIA [41]) treat security as a principal design goal, and we expect it will be easier to explore *generic* (without application-layer support) traffic analysis resistance in this context.

This is an open ended exploratory project. We will limit its scope to document publication and retrieval, primarily because this application is more latency-tolerant than anything related to real-time conversations (chat, VoIP, etc.) and secondarily because this is the principal battlefield for online censorship and surveillance. Note that we are not giving up much; the existing capabilities of the "Web platform" already permit construction of applications with nearly all logic on the client side, the server only acting as a distribution point for static data, and planned additional features will only make this easier. In fact, moving logic to the client automatically reduces users' exposure to traffic analysis: for instance, it would not be nearly as easy to extract information from a tax-preparation application if its logic was transmitted to the client all at once, and only the finished paperwork was sent back, instead of a long sequence of questions and answers being sent over the network one at a time.

Publish-retrieve systems with some degree of anonymity, censorship resistance, and/or surveillance resistance have been proposed before; designs of note include Free Haven [17], Tangler [38], and the Eternity Service [3, 5]. So far none has achieved wide deployment, and some have been found to have serious flaws [27]. Distributed caches can reveal retrieval history; even when caches are not supposed to be able to learn decryption keys for the content they hold, as in Serjantov [35], keyword searches for undesirable content can finger caches that hold it, and thus the users that might have retrieved it.

However, the basic notion of distributed storage for self-validating, content-addressed material seems sound, and presents a number of desirable affordances: The threat models and designs in this space typically do include explicit consideration of retrieval anonymity and censorship and surveillance resistance. Opportunistic retrieval and caching (or validation) offer a degree of unobservability, by offering a node plausible deniability as to whether any given data object was retrieved by a human. Tangler in particular suggests a way toward genuinely oblivious caching, i.e. the presence of a particular encrypted blob in a cache need not reveal that a particular document has ever been retrieved through that cache (even if the adversary knows *some* of the documents that that blob can produce). Finally, although it is not directly related to the question of traffic analysis, these systems all build on a notion of signing keys for content, which could supersede the Web's creaky scheme of certificate authorities (as long as some method can be found to name these keys which isn't inscrutable to humans).

3 Social Considerations

We are also concerned with what the "new" Web, providing (let us suppose) true anonymity for readers, strong pseudonymity for authors, and strong guarantees that content once published cannot be removed, will look like as a social phenomenon. In particular, the *present* online culture has tendencies toward groupthink, abuse of outsiders, and general hostility toward anyone who looks like a tempting target. One might argue that "baking in" masks for everyone to hide behind will only make all of this worse.

We first observe that no technical measure can hope to distinguish "legitimate" political speech from harassment of private citizens from straight-up trolling with certainty. Merely deciding whether or not a document expresses a negative opinion, in the general case, is an AI-complete problem.[2] Creating a network that would somehow prevent online abuse without also chilling political speech acts is *harder* than AI-complete: even if we had the necessary AI, who would get to decide what policy it would apply? Consider that one person's idea of a legitimate political cartoon may be another person's idea of *lèse-majesté* or blasphemy.

However, we do not think the situation is hopeless. Spam filters are not perfect, but do a good enough job to be useful, especially in conjunction with human moderation. The same statistical and heuristic techniques could be applied to filter out trolls and bullies. Further, content filters and moderation are not intrinsically unethical; what is problematic from a freedom-of-speech perspective is when they are imposed on individuals who would prefer not to have them, or when they are used to punish the authors of controversial content. We suggest that the application layer could offer different views of the same site to all visitors, allowing them to pick their desired level of curation. Exactly how this would work is beyond the scope of the present proposal, but we are interested in developing it further, down the road.

References

[1] N. Aase, J. R. Crandall, Á. Díaz, J. Knockel, J. O. Molinero, J. Saia, D. Wallach, and T. Zhu. "Whiskey, Weed, and Wukan on the World Wide Web: On Measuring Censors' Resources

[2] By analogy to NP-completeness, "AI-complete" refers to problems that, if solved in software, would imply that a human-equivalent artificial intelligence had been created.

and Motivations." In: *Free and Open Communications on the Internet*. 2012. usenix: foci12/aase. URL: https://www.usenix.org/system/files/conference/foci12/foci12-final17.pdf.

[2] S. Ackerman. "Lavabit email service abruptly shut down citing government interference." In: *The Guardian* (Aug. 2013). URL: http://www.theguardian.com/technology/2013/aug/08/lavabit-email-shut-down-edward-snowden.

[3] R. Anderson. "The Eternity Service." In: *Pragocrypt*. 1996. URL: http://www.cl.cam.ac.uk/users/rja14/eternity/eternity.html.

[4] M. Backes, G. Doychev, and B. Köpf. "Preventing Side-channel Leaks in Web Traffic: A Formal Approach." In: *Network and Distributed System Security Symposium*. Internet Society, Feb. 2013. anonbib: ndss13-website-fingerprinting.

[5] T. Benes. "The Strong Eternity Service." In: *Information Hiding Workshop*. Vol. 2137. Lecture Notes on Computer Science. 2001. URL: http://freehaven.net/anonbib/papers/strong-eternity.pdf.

[6] X. Cai, X. C. Zhang, B. Joshi, and R. Johnson. "Touching from a Distance: Website Fingerprinting Attacks and Defenses." In: *Computer and Communications Security*. 2012. URL: http://www.cs.stonybrook.edu/~xcai/fp.pdf.

[7] C. Castelluccia, E. De Cristofaro, and D. Perito. "Private Information Disclosure from Web Searches (The case of Google Web History)." In: (2010), pp. 38–55. arXiv: 1003.3242 [cs.CR].

[8] P. Chapman and D. Evans. "Automated Black-Box Detection of Side-Channel Vulnerabilities in Web Applications." In: *Computer and communications security*. 2011, pp. 263–274. URL: http://qosbox.cs.virginia.edu/~evans/pubs/ccs2011/sca-packaged.pdf.

[9] D. Chaum. "Untraceable electronic mail, return addresses, and digital pseudonyms." In: *Communications of the ACM* 24.2 (1981). DOI: 10.1145/358549.358563.

[10] S. Chen, R. Wang, X. Wang, and K. Zhang. "Side-channel leaks in web applications: A reality today, a challenge tomorrow." In: *Symposium on Security and Privacy*. IEEE. 2010, pp. 191–206. URL: http://research.microsoft.com/pubs/119060/WebAppSideChannel-final.pdf.

[11] H. Cheng and R. Avnur. *Traffic Analysis of SSL Encrypted Web Browsing*. Tech. rep. University of California, Berkeley, 1998. URL: http://www.eecs.berkeley.edu/~daw/teaching/cs261-f98/projects/final-reports/ronathan-heyning.ps.

[12] S. E. Coull, M. P. Collins, C. V. Wright, F. Monrose, and M. K. Reiter. "On Web Browsing Privacy in Anonymized NetFlows." In: *USENIX Security Symposium*. 2007, pp. 339–352. usenix: sec07/coull.

[13] G. Danezis. "Statistical Disclosure Attacks: Traffic Confirmation in Open Environments." In: *Security and Privacy in the Age of Uncertainty*. 2003, pp. 421–426. anonbib: statistical-disclosure.

[14] G. Danezis. "The Traffic Analysis of Continuous-Time Mixes." In: *Privacy Enhancing Technologies*. Vol. 3424. Lecture Notes in Computer Science. May 2004, pp. 35–50. anonbib: danezis-pet2004.

[15] G. Danezis, C. Díaz, and C. Troncoso. "Two-Sided Statistical Disclosure Attack." In: *Privacy Enhancing Technologies*. 2007, pp. 30–44. anonbib: danezis-pet2007.

[16] G. Danezis and A. Serjantov. "Statistical Disclosure or Intersection Attacks on Anonymity Systems." In: *Information Hiding*. 2005, pp. 293–308. anonbib-psgz: DanSer04.

[17] R. Dingledine, M. J. Freedman, and D. Molnar. "The Free Haven Project: Distributed Anonymous Storage Service." In: *Designing Privacy Enhancing Technologies*. Vol. 2009. Lecture Notes in Computer Science. 2000. URL: http://freehaven.net/doc/berk/freehaven-berk.ps.

[18] R. Dingledine, N. Mathewson, and P. Syverson. "Tor: The Second-Generation Onion Router." In: *USENIX Security Symposium*. 2004, pp. 303–320. usenix: sec04/dingledine.

[19] K. P. Dyer, S. E. Coull, T. Ristenpart, and T. Shrimpton. "Peek-a-Boo, I Still See You: Why Efficient Traffic Analysis Countermeasures Fail." In: *Symposium on Security and Privacy*. IEEE. 2012. URL: http://kpdyer.com/publications/oakland2012.pdf.

[20] E. W. Felten. *ACLU v. Clapper. (Felten Decl.)* Southern District of New York, 1:2013cv03994. 2013. URL: https://ia601803.us.archive.org/22/items/gov.uscourts.nysd.413072/gov.uscourts.nysd.413072.27.0.pdf.

[21] Y. Gluck, N. Harris, and A. Prado. "BREACH: Reviving the CRIME Attack." In: *Black Hat*. 2013. URL: https://www.blackhat.com/us-13/archives.html#Prado.

[22] G. Greenwald et al., eds. *The NSA Files*. The Guardian, 2013–. URL: http://www.theguardian.com/world/the-nsa-files.

[23] D. Herrmann, C. Gerber, C. Banse, and H. Federrath. "Analyzing Characteristic Host Access Patterns for Re-identification of Web User Sessions." In: *Information Security Technology for Applications*. Ed. by T. Aura, K. Järvinen, and K. Nyberg. Vol. 7127. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 136–154. ISBN: 978-3-642-27936-2. DOI: 10.1007/978-3-642-27937-9_10. URL: http://epub.uni-regensburg.de/18731/1/nordsec_10_7_final.pdf.

[24] N. Hopper, L. von Ahn, and J. Langford. "Provably secure steganography." In: *IEEE Transactions on Computers* 58.5 (2009), pp. 662–676. citeseer: 10.1.1.153.7785.

[25] W. Jia, F. P. Tso, Z. Ling, X. Fu, D. Xuan, and W. Yu. "Blind detection of spread spectrum flow watermarks." In: *Security and Communication Networks* 6.3 (2013), pp. 257–274. ISSN: 1939-0122. DOI: 10.1002/sec.540. citeseer: 10.1.1.156.4883.

[26] D. Kahn. *The Codebreakers: The Story of Secret Writing*. New York: Macmillan, 1967.

[27] D. Kügler. "An Analysis of GNUnet and the Implications for Anonymous, Censorship-Resistant Networks." In: *Privacy Enhancing Technologies*. Vol. 2760. Lecture Notes on Computer Science. Mar. 2003, pp. 161–176. URL: http://www.ovmj.org/GNUnet/papers/GNUnet_pet.pdf.

[28] L. Mather and E. Oswald. "Pinpointing side-channel information leaks in web applications." In: *Journal of Cryptographic Engineering* 2 (3 2012), pp. 161–177. DOI: 10.1007/s13389-012-0036-0.

[29] S. J. Murdoch and P. Zieliński. "Sampled Traffic Analysis by Internet-Exchange-Level Adversaries." In: *Privacy Enhancing Technologies*. Ed. by N. Borisov and P. Golle. Lecture Notes in Computer Science. Ottawa, Canada: Springer, June 2007. anonbib: murdoch-pet2007.

[30] R. Newman. *The Church of Scientology vs. anon.penet.fi*. Web page. 1996. URL: http://www.spaink.net/cos/rnewman/anon/penet.html.

[31] A. Pfitzmann and M. Hansen. *A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management*. Version 0.34. 2010. URL: http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf.

[32] R. Pickholtz, D. Schilling, and L. Milstein. "Theory of Spread-Spectrum Communications—A Tutorial." In: *IEEE Transactions on Communications* 30.5 (1982), pp. 855–884. citeseer: 10.1.1.114.208.

[33] A. Pironti, P.-Y. Strub, and K. Bhargavan. *Identifying Website Users by TLS Traffic Analysis: New Attacks and Effective Countermeasures*. Tech. rep. RR-8067. INRIA, 2012. URL: http://hal.inria.fr/hal-00732449.

[34] J. Rizzo and T. Duong. "The CRIME attack." In: *ekoparty security conference*. 2012. URL: http://www.ekoparty.org/archive/2012/CRIME_ekoparty2012.pdf.

[35] A. Serjantov. "Anonymizing censorship resistant systems." In: *Peer-to-Peer Systems*. Springer, 2002, pp. 111–120.

[36] V. Shmatikov and M.-H. Wang. "Timing Analysis in Low-Latency Mix Networks: Attacks and Defenses." In: *ESORICS*. Sept. 2006. anonbib: ShWa-Timing06.

[37] R. Singel. “Encrypted E-Mail Company Hushmail Spills to Feds.” In: *Wired: Threat Level* (Nov. 2007). URL: http://www.wired.com/threatlevel/2007/11/encrypted-e-mai/.

[38] M. Waldman and D. Mazières. “Tangler: a censorship-resistant publishing system based on document entanglements.” In: *Computer and Communications Security*. 2001, pp. 126–135. URL: http://www.cs.nyu.edu/~waldman/tangler.ps.

[39] M. Wright, M. Adler, B. N. Levine, and C. Shields. “An Analysis of the Degradation of Anonymous Protocols.” In: *Network and Distributed Security Symposium*. 2002. anonbib: wright02. URL: http://www.cs.umass.edu/~mwright/papers/wright-degrade.pdf.

[40] M. Wright, M. Adler, B. N. Levine, and C. Shields. “Defending Anonymous Communication Against Passive Logging Attacks.” In: *Symposium on Security and Privacy*. 2003. anonbib: wright03. URL: http://www.cs.umass.edu/~mwright/papers/wright-passive.pdf.

[41] *eXpressive Internet Architecture*. Web site. 2013. URL: https://www.cs.cmu.edu/~xia/.

[42] W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao. “DSSS-based flow marking technique for invisible traceback.” In: *Symposium on Security and Privacy*. IEEE. 2007, pp. 18–32. citeseer: 10.1.1.117.6637.

[43] K. Zhang, Z. Li, R. Wang, X. Wang, and S. Chen. “Sidebuster: Automated Detection and Quantification of Side-Channel Leaks in Web Application Development.” In: *Computer and Communications Security*. ACM. 2010, pp. 595–606. URL: http://www.cs.indiana.edu/~lizho/sidebuster-final.pdf.

[44] L. Øverlier and P. Syverson. “Locating hidden servers.” In: *Symposium on Security and Privacy*. IEEE. 2006. anonbib: hs-attack06.