Better Not to Know

Polemic Computation, and Ecological Costs in the SHA1 Collision Compute

Nick Merrill
BioSENSE, UC Berkeley School of Information
Berkeley, California, USA
ffff@berkeley.edu

ACM Reference format:
Nick Merrill. 2017. Better Not to Know
Polemic Computation, and Ecological Costs in the SHA1 Collision Compute. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 4 pages.
DOI: 10.1145/nnnnnnn.nnnnnnn

> I insist on the fact that there is generally no growth but only a luxurious squandering of energy in every form!
>
> Georges Batailles, *The Accursed Share*

1 TODO INTROUDCTION

some large computes, of course, are for the best genome, protein folding, etc… and this compute was possibly for the best… some SSL certs use SHA1

but …. my question is whether or not they should perform it, vs "proving" the attack in theory, given the ecological costs (energy, CO2) and the opportunity costs (what else could they could have computed, e.g. protein folding)? why compute at all?

polemic aims (even important ones) weigh political and social goals against ecological risks DRILL IN – POLEMIC AIMS FOR **SPECIFIC** STAKHEOLDERS….. MORE ABOUT WHY THIS IS ABOUT "LIMITS" PER SE

This paper reads the SHA1 collision compute, and the various sociotechnical entanglements that motivated it to be performed (rather than simply discussed) (Section 2), to motivate a throy forwards of *polemic computation* (Section 3), brief definition. I discuss the weighing of polemic rewards against the ecological risks in (Section 5).

using this example of a MEDIUM sized compute highlights BROADER tensions about when to compute and when not to especially in the case of EXTREMELY large computes…. always unclear risk/reward, e.g. with training neural nets start to raise questions about what we can we do to hedge our risks - in time and captial, but also in the environment? we will only have more things to compute, and more things to compute them with, but how to select, how to use restraint?

Conference'17, Washington, DC, USA

DOI: 10.1145/nnnnnnn.nnnnnnn

2 BACKGROUND

Before discussing Google's large compute in depth, this section gives some background on SHA1, and cryptographic hash functions in general. Cryptographic hash functions are "one way" functions: they take some data, and produce some new data, such that the original data cannot be recovered from the new data. The output of the hash function is simply called a *hash*.

SHA1 is one cryptographic hash function, designed by the NSA in the early 1990s. For some block of any-sized data, SHA1 produces a 40-digit string of characters. It is used in most version control applications to refer uniquely to files. SHA1 may also be used to check for corrupted files. Crucially, as we will discuss in Section 3, SHA1 is also used in security-oriented protocols such as SSH/TLS.

2.1 SHA1 collisions

The hashes output by SHA-1 are typically 40 digits, regardless of the size of the input data. Crucially, hashes *should* relate uniquely to input data: two different inputs should never produce the same hash (even though hashes are much smaller than the original data). A *collision* refers to the breakdown of this property, in which two different input data produce identical hashes.

Collisions break several common uses of SHA1. (Amusingly, a test that captured Google's collision test broke the version control system for the Webkit browser engine [4]). In the case of SSL/TLS, the protocol for encrypted and authenticated communication on the web, SHA1 collisions have more severe consequences. Section 3 will return to TLS vulnerability in more detail.

2.2 SHA1 collisions in theory

In discussing the safey of particular hash functions, two questions must be asked: (1) how long would it take to find a collision by brute force?, and (2) is there any algorithm that allows us to find a collision faster than the brute force method? For the brute force method, the odds of finding a SHA-1 collision by chance are one/2^{80} [7]. In general, the security of this brute-force attack is judged relative to the outer edge of high-end hardware, and hash functions are expected to be retired in time, as computers grow more powerful. However, this 2^{80} space of possibilities in the search for a collision is not considered feasible, so SHA-1 appears safe.

In 2005, however, Wang, Yin & Yu found an algorithm to produce SHA-1 collisions in under 2^{69} calculations (about 2,0000 times faster than the brute force approach) [14]. (It is worth noting that other work had suggested possible weaknesses of SHA-1 earlier [2]). While such a compute was, at the time, outside the limits of even powerful adversaries, the result caused concern among cryptographers [7]. By 2011, a 2^{61} calculation attack was discovered

[10], and by the mid 2010s, the developers of most major browsers had announced plans to stop accepting SHA-1 SSL certificates [].

3 GOOGLE'S WORK: A PRACTICAL SHA1 COLLISION

two pdfs with different content, but identical SHA-1 hashes. [12]. the study in question performed a SHA collision in $2^{63.1}$ computations, and released the source code for replicating the attack [11]. Compared to the 2^{61} theoretical attack, the practical attack took a bit longer due to the communication overhead required to coordinate computations across several datacenters, and due to the relative inefficiency of using GPUs rather than CPUs.

In practice, the computation required to produce the SHA1 collision required 6,500 years of CPU time and 110 years of GPU time. While this number certainly sounds high, 600,000 cores, each running two threads, could take only two days of compute time.

Of course, time is not the only cost to consider. Computation is material, physically instiantiated, and has physical, ecological consequences. What did the SHA1 compute cost in other terms? some energy costs estimated in the paper apparently calculate cost relative to google research operating budget, alphabet budget The cost of such a compute on Cray supercomputers would be on the order of a million dollars [6] (though such estimates might vary widely in either direction from system to system).

This section gave background on SHA1 collisions, and gave context for the costs (in time and energy) of the SHA1 collision compute. The following section details possible explanations for why the computation was performed in practice, rather than simply discussed in theory.

4 EXPLAINING WHY THE COMPUTE HAPPENED

Since a theoretical result already existed showing a SHA1 collision was possible, one might rightly wonder why researchers would go through a great deal of time and effort (not to mention a great deal of expense, both monetary and ecological) to produce artifacts of no practical purpose (different PDFs with identical checksums). What are the possible benefits?

In this section argues that the SHA-1 collision compute had essentially polemic goals. It was performed not to know a particular answer (as the PDFs themselves are not useful as artifacts), but to know that such an answer *has* been found, as opposed to *can* be found. go on, entangled in various social and economic.....especially in the case of SSL certs

4.1 Practice versus theory

One explanation for performing the result is academic. The computation ended up being more difficult than theoretical results indicated due to the storage and communication requirements necessary to perform the work across multiple datacenters. The authors also count some of the energy needs for this attack, which is relevant in knowing how realistic the threat is in practice.

All cryptography can be broken with sufficient computational time. Cybersecurity practice depends on navigating the costs and benefits for adversaries, and their presumed resourcefulness This result shows that a powerful attack (such as Google) can indeed break SHA1 with some knowable resources. Surely, if Google can perform such an attack, a government actor could do so as well. One question raised by the polemics of this compute is, "do SHA-1 users have assets worth at least as much as the cost of this compute?"

Of course, given that SHA1 has already been widely deprecated, this explanation does not answer why such an academic exercise was considered necessary. summarize next section.

4.2 Message to users of SHA1

Some users of SHA1 did not care much about the demonstrated attack. Linus Torvalds, developer of the Git version control software (which relies on SHA1 to refer to files), reported no immediate concern. "Do we want to migrate to another hash? Yes. Is it 'game over' for SHA1 like people want to say? Probably not." [13].

SHA1 is also used in the issuance of (especially older) SSL certificates. (SSL certificates provide a token of the authenticity of a user's connection to a webpage, and encrypts data end-to-end). This practical result showed that soembody with the power to perform a SHA1 collision could now make a fake certificate for a website with that uses SHA1 for nerits TLS. Such a false certificate could be used to convince a victim that they are communicating with a given website, when in fact they are communicating with the attacker.

SSL certificates are issued by Certificate Authorities (CAs), which in theory abide by regulations set by the CA/Browser Forum, a standards-setting body. Here unravels a more complex story of regulation and standards bodies, as well as stakeholders for whom a change away from SHA-1 could incur significant monetary costs. The following sections examine the impact of this attack on both CAs, and browser developers.

4.2.1 Certificate authorities. First, the SHA-1 attack can be mediated entirely by by replacing old SHA1 certificates with newer ones using SHA-2 or SHA-3. Second, CAs that abide by CA/Browswer Forum rules are already forbiddedn from issuing SHA-1 certificates. (They are additionally required to insert at least 64 bits of randomness, in an effort to mitigate devistating effects from future cryptographic breaks) [].

However, Since CAs are decentralized, and since SSL issuees (website administrators) do not routinely check issued SSL certificates for these properties, enforcing these regulations is a perenial challenge for the CA/Browser Forum. It is not clear that CAs were abiding by either of these rules.

One explanation here is that google wanted people to wise up and stop using SHA-1 for real, or else

4.2.2 ***TODO*** *Browser press.* Alongside the issue of enforcing proper security practices on a decentralized system of certificate authorities, there is a separate ecosystem of browser producers. While browser production is also decentralized, it is less so than CAs (what % of people use browsers? chart here could help? maybe a figure of CAs as well). lots of browsers have stopped accepting SHA-1: [3, 5]. (between windows, chrome, firefox some % percent of web browsers would have no longer accepted SHA-1 signed SSL certificates, even if the compute never took place.....

however, the pressroom story.....gotten flack at a very high level, browsers need their users to feel secure, or people will not use the web as much!!!!!! „„„ and gotten flack for it„„„„„„

so, another explanation is that this thing helps their PR by bolstering their decision, making it seem more reasonable or wahtever.

4.3 TODO Extravegance

This collision demonstrated not only the considerable resources required to exploit []'s theoretical result, but the vast resources that Google must have, if it is able to spend so heavily on a project with essentially polemic aims. Alternatively, the computation had the ulterior polemic aim of demonstrating Google's resources. By Bataille's theory of consumption [], some share of all economic activity must be spent without gain expand quite a bit here.....

5 TODO THE POLEMICS OF ACTUALLY DOING

summarize last section

summarize this section

5.1 Defining polemic computation

This paper defines *polemic computation* as a computation enacted (rather than discussed) in order to forward an argument or ideology. Crucially, computations are material artifacts, produced in time and energy []. Additionally, their performance or enactment requires specialized technical expertise in the form of labor. Polemic computes are at once feats and artifacts, which act as an agent [] in technosocial debates.

The following sections relate this theory of polemic computations to other theories of charismatic technology and critical design, highlighting the relevant differences between our theories and these.

5.2 Charismatic technology

Indeed, polemic computation can be said to "work" in part because it is animated by ideological frameworks. In the case of the SHA-1 computation, ideals that web communications *should* be private and authenticated very much animate the particular computations that occurred. These ideals become especially clear when one examines the motivations for actually performing the compute, even though they were already discussed in theory.

Polemic computation draws srongly to Ames' theory of *charismatic technology* []. Drawing on actor-network theory, charismatic technology would ascribe the very artifact of the computation (a material artifact produced by material means [dorish]) agency in the technosocial discourses around privacy and security. Indeed, much like in Ames case of the One Laptop Per Child project, polemic computation aims to change behavior and beliefs among specific stakeholders in specific

However, in contrast to charismatic technology, polemic computation centers the material act of computing as a *feat* with costs in time and energy. In energy, computation expends valuable and scarce ecological resources []. In time and energy, computational incurs opportunity costs, through answers that could have been computed but were not. Rather than computing answers, polemic computation uses the material feat of expenditutre to work as an agent in technosocial discourse.

5.3 Critical design

Another strand of research that explicitly centers the agency of technological artifacts is critical design [dunn n rabie i guess]. Critical design seeks to harness the agency of technical artifacts to challenge assumptions or surface lurking cultural narratives. In many ways, polemic computation serves as a critical artifact. The SHA-1 collision compute, for example, called out the poor security practices of many certificate authorities. Specifically, the material production of the computation, combined with its almost satirical nature (the compute produced PDFs), acted to *define* what is and is not a poor security practice for certificate authorities. Much in the tradition of critical design used its material power [bennett] along with a touch of humor to enter into technosocial debates and imaginaries.

6 ECOLOGICAL RISKS, POLEMIC REWARDS

The prior section outlined explanations for why the SHA1 compute was performed, and proposed a theory of polemic computation typifying such explanations. A separate question that I have not yet addressed is whether or not the compute *should* have been performed, given the ecological costs (energy and CO2), and the opportunity costs (what else could they could have computed instead, e.g. protein folding).

More generally, in the case of computations with polemic aims, how do we decide when to compute? How can we weigh costs (of all sorts) against the potential (polemic) benefits? This question could be framed from both an ethical perspective, and from an econometric one.

In this section, I outline a few kinds of large compute projects, highlighting ways in which they could be considered polemic, and surfacing the field of risks (and rewards) associated with each. In general, future work should explore the space of risks and rewards associated with polemic computes from a variety of ethical, legal and economic standpoints.

6.1 Volunteer distributed computing projects

Some projects have aimed to perform large computations by distributing the work across multiple machines, particularly commodity hardware supplied by volunteers. A popular platform, BOINC (Berkeley Open Infrastructure for Network Computing) allows projects to utilize a vast network of volunteers' computing time, for example, when their laptop is idle, as a screensaver [1]. The power of this approach lies in its ability to scale "horizontally," across a wide variety of readily-accessible (and widely deployed) machines.

However, individual machines may not be as efficient in power as large-scale server farms. Additional costs in energy are incurred by added network transmissions, and the generally lower power-efficiency of commodity devices. These projects reduce capital overhead for those running the compute, but may exarcerbate ecological risks.

Future work might examine volunteer computing projects through the lens of polemic computation. Projects like SETI (Search for Extraterrestrial Intelligence at Home), which have users perform fast fourier transforms on billions of hours of radio recordings, serve as much to engage in discourses around science and the public as they do to produce useable data [1]. The computational work (and

associated costs) might be fruitfully examined to other distributed projects, such as protein folding.

6.2 Web applications as supercomputation

Web applications share some properties with the volunteer distributed computing applications mentioned above. Much computation is offloaded onto commdity clients, such as mobile phone apps or web browsers. Consumers of these applications trade their computational time, and electricity, in exchange for the service. Consider netflix, which retains a centralized system of indexing and content delivery, but offloads to consumers the processing associated with watching videos (downloading videos, along with decrypting the digital rights management, decoding the video format, and finally playing the video and audio).

Future work might exammine the motivations for architectural decisions in web applications through the polemics around Web 2.0 [8], examining how discourse around "thin clients" and "the cloud" interact with technical constraints to influence decisions in where processing takes place. Such polemic decisions may have real ecological consequences.

6.3 Rise of machine learning

Some work in machine learning blurs the line between polemic intent and answer-finding. Image recognition benchmarks provide one example of this phenomenon: while a good image recognition algorithm certainly *can* have intrinsic value in other domains (e..g in transfer learning [**?**]), the production of such an algorithm is often incidental to the production of the benchmark. Benchmarks serve to mark or legitmize the algorithm's architecture (especially in the case of neural nets) for the classification problem.

Meanwhile, contemporary machine learning techniques, especially the training of neural nets, require a tremendous amount of computation, and therefore a large expenditure of energy. Thus, when training algorithms in a computationally complex way, we must ask questions about the costs (and motivations) for doing so. Future work could raise questions about the polemics involved with particular attempts to train deep learning algorithms, examining their ecological costs against the sociotechnical goals of performance in particular competitions, or against particular benchmarks [9].

7 TODO CONCLUSION

sequencing the genome, as much to excite a field as anything else more ambitious with mapping human brain – what are the results I argue many supercompute projects have this quality

computing ever-more digits of pi for example; dubious empirical benefit (at least for now/until it doesnt; with math especially, one never knows).

Polemic rewards Ecological risks _Enforcing a balance

8 ACKNOWLEDGEMENTS

Morgan Aimes, Nick Doty, Anette Greiner, Sebastian Benthall

REFERENCES

[1] David P Anderson. 2004. Public Computing : Reconnecting People to Science. *Knowledge Creation Diffusion Utilization* (2004), 1–6. http://boinc.berkeley.edu/boinc2.pdf

[2] Eli Biham, Rafi Chen, Antoine Joux, Patrick Carribault, Christophe Lemuet, and William Jalby. 2005. *Collisions of SHA-0 and Reduced SHA-1.* Springer Berlin Heidelberg, Berlin, Heidelberg, 36–57. DOI: http://dx.doi.org/10.1007/11426639_3

[3] Ryan et al Sleevi. 2014. Intent to Deprecate: SHA-1 certificates. (2014). https://groups.google.com/a/chromium.org/forum/

[4] Antti Koivisto. 2017. Bug 168774 - Add a test verifying cache deduplication is not sensitive to SHA1 collision attack. (2017). https://bugs.webkit.org/show

[5] Mozilla. 2017. CA:Problematic Practices. (2017). https://wiki.mozilla.org/CA:Problematic

[6] Greg Pautsch, Duncan Roweth, and Scott Schroeder. 2016. *The Cray® XCTM Supercomputer Series: Energy-Efficient Computing.* Technical Report. Cray, Inc. 23 pages.

[7] Bruce Schneier. 2005. Cryptanalysis of SHA-1. (2005). https://www.schneier.com/blog/archives/2005/02/cryptanalysis

[8] Trebor Scholz. 2008. Market ideology and the myths of web 2.0. *First Monday* 13, 3 (2008). DOI: http://dx.doi.org/10.1007/s13398-014-0173-7.2 arXiv:arXiv:1011.1669v3

[9] M. Six Silberman. 2015. Information systems for the age of consequences. *First Monday* 20, 8 (2015), 1–1.

[10] Marc Stevens. 2013. New collision attacks on SHA-1 based on optimal joint local-collision analysis. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 7881 LNCS. 245–261. DOI: http://dx.doi.org/10.1007/978-3-642-38348-9_15

[11] Marc Stevens. 2017. cr-marcstevens/sha1collisiondetection. (2017). https://github.com/cr-marcstevens/sha1collisiondetection

[12] Marc Stevens, Elie Bursztein, Pierre Karpman, Ange Albertini, and Yarik Markov. 2017. The first collision for full SHA-1. (2017). https://shattered.it/static/shattered.pdf

[13] Linus Torvalds. 2017. Re: SHA1 collisions found. (2017). http://marc.info/?l=git

[14] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. 2005. *Finding Collisions in the Full SHA-1.* Springer Berlin Heidelberg, Berlin, Heidelberg, 17–36. DOI: http://dx.doi.org/10.1007/11535218_2