Better Not to Know

The SHA1 Collision & the Limits of Polemic Computation

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

ABSTRACT

In February of 2017, Google announced the first SHA1 collision. Using over nine quintillion computations (over 6,500 years of compute time), a group of academic and industry researchers produced two different PDF files with identical SHA1 checksums. But why? After all, SHA1 had already been deprecated by numerous regulatory and advisory bodies. This paper uses the SHA1 collision compute as a site for surfacing the space of ecological risks, and sociotechnical rewards, associated with the performance of large computes. I forward a theory of *polemic computation*, in which the feat of expending material (ecological, labor, and opportunity) costs exert agency in particular sociotechnical discourses. I conclude with challenges for future work in balancing the costs and rewards of computes with polemic aims.

CCS CONCEPTS

• **Applied computing → Computers in other domains**; • **Human-centered computing** → *HCI theory, concepts and models*;

KEYWORDS

theory, limits, polemics, charisma

ACM Reference format:

Nick Merrill. 2017. Better Not to Know
The SHA1 Collision & the Limits of Polemic Computation.

In *Proceedings of ACM Limits Workshop, Santa Barbara, California USA, June 2017 (LIMITS '17),* 6 pages.
DOI: 10.475/123_4

> 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 INTROUDCTION

From protein folding to the discovery of novel drugs, some large computes are for the best [2]. But what about computing ever-increasing digits of pi? What about finding two PDFs with different content, but identical checksums? Are such computes worth the ecological costs, and the opportunity costs, compared to other things one could compute?

This paper reads the SHA1 collision compute (Section 2), and the various sociotechnical entanglements that motivated it to be performed (rather than simply discussed) (Section 3), to motivate a theory of *polemic computation* (Section 4), computations that work primarily to enter into sociotechnical discourses. In the case of the SHA1 compute, researchers took on a sizeable monetary and environmental cost to enforce a point of view among specific stakeholders in the (social, political) economy of SSL certificates. In so doing, the compute's aims are animated by larger narratives of authenticity, privacy and security on the Internet.

Polemic computes, even important ones, weigh political and social goals against ecological (and other) costs. This paper aims to begin questions about how polemic computes can and should work within limits. This paper does not offer a rigorous theory of how polemics might be balanced against limits (of time, energy, opportunity); rather I aim to surface questions about the limits of how far computation *should* go to prove our point. I conclude with how this debate might carry forward in various applications ranging from distributed computing to web applications and machine learning (Section 6).

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 I 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 SHA1 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

LIMITS '17, Santa Barbara, California USA
 123-4567-24-567/08/06. . . $15.00
DOI: 10.475/123_4

system for the WebKit browser engine [10]). In the case of SSL/TLS, the protocol for encrypted and authenticated communication on the web, SHA1 collisions have more severe consequences. Section 5 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 SHA1 collision by chance are one/2^{80} [14]. 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 SHA1 appears safe.

In 2005, however, Wang, Yin & Yu found an algorithm to produce SHA1 collisions in under 2^{69} calculations (about 2,0000 times faster than the brute force approach) [22]. (It is worth noting that other work had suggested possible weaknesses of SHA1 earlier [4]). While such a compute was, at the time, outside the limits of even powerful adversaries, the result caused concern among cryptographers [14]. By 2011, a 2^{61} calculation attack was discovered [18], and by the mid 2010s, the developers of most major browsers had announced plans to stop accepting SHA1 SSL certificates [8, 11].

3 PERFORMING A COLLISION

The study in question here produced two PDFs with different content, but identical SHA1 hashes [20]. the study in question performed a SHA collision in $2^{63.1}$ computations, and released the source code for replicating the attack [19].

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 instantiated, and has ecological consequences. Beyond monetary cost, such large computations have very real costs in energy. Since the implementation details of the infrastructures used for the large collision compute are not entirely knowable from the paper, it is difficult to estimate this energy cost. As a rough point of comparison, the monetary cost of such a compute on Cray supercomputers would be on the order of one million USD (though such estimates might vary widely in either direction from system to system) [12]. In any case, such a figure is a tiny sliver of Alphabet Inc.'s 90 billion USD revenue in 2016.

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 I argue that the SHA1 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. I argue that the performance of this collision compute was necessarily entangled in a particular sociotechnical discourse, and aimed to change opinions and behavior among specific groups of stakeholders. This section focuses in particular on those involved in the ecosystem of SSL certificates: browsers, webmasters, and the certificate authorities (CAs) tasked with generating certificates.

4.1 Practice versus theory

Before progressing onto a discussion of this compute on the ecosystem of SSL certificates, we must briefly argue for why an argument of academic interest does not sufficiently explain why this compute was performed, rather than simply discussed.

The computation here 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 collision compute reveals details relevant to knowing how realistic the threat is in practice. Of course, given that SHA1 has already been widely deprecated, this explanation does not in itself answer why such an exercise was considered necessary. After all, one would not need to know the cost in practice of such an exercise without some reason.

Indeed, one reason, aside from the particular answer computed, is that the compute raises a question and challenge to users of SHA1: "Do SHA1 users have assets worth at least as much as the cost of this compute?" All cryptography can be broken with sufficient computational time. This result shows that a powerful attack (such as Google) can indeed break SHA1 with some knowable resources. And, surely, if Google can perform such an attack, a government actor could do so as well.

Of course, 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." [21]. The following section explains the performance of this collision in the context of an application in which stakes are potentially much higher: the issuance of SSL certificates, some of which rely on SHA1 to provide cryptographic guarantees.

4.2 SHA1 and SSL Certificates

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 someone with the power to perform a SHA1

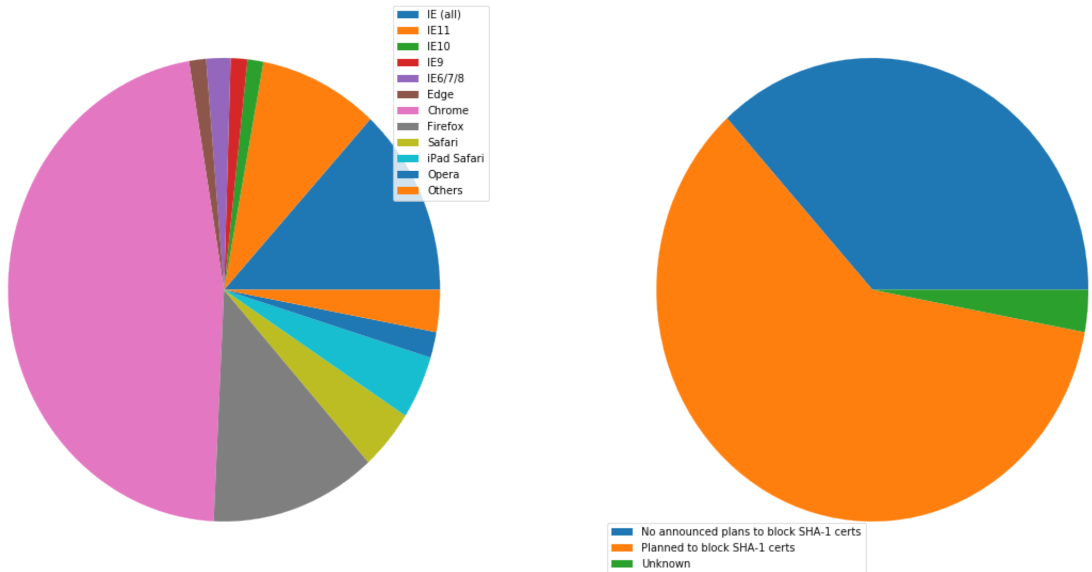

Figure 1: TODO Explain how many browsers would have ignored SHA1s anwyway, even if this compute never happened.

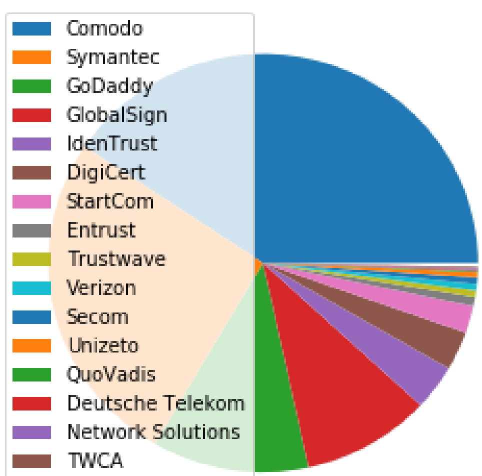

Figure 2: TODO Explain how many browsers would have ignored SHA1s anwyway, even if this compute never happened.

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 SHA1 could incur significant monetary costs. The following sections examine the polemic impact of this attack on both CAs, and browser developers.

4.2.1 Certificate authorities. First, the SHA1 attack can be mediated entirely by replacing old SHA1 certificates with newer ones using SHA-2 or SHA-3. Second, CAs that abide by CA/Browser Forum rules are already forbidden from issuing SHA1 certificates. (They are additionally required to insert at least 64 bits of randomness, in an effort to mitigate devastating effects from future cryptographic breaks) [20].

However, Since CAs are decentralized, and since SSL issues (website administrators) do not routinely check issued SSL certificates for these properties, enforcing these regulations is a perennial challenge for the CA/Browser Forum. It is not clear that CAs were abiding by either of these rules. There exists a long tail of small certificate authorities (Figure 1), in comparison to the relative centralization of browser production (Figure 2) [13]. Assuming they were not, one explanation for performing this compute is that doing so would encourage CAs (and webmasters) to take more seriously the threat posed by SHA1, putting some real pressure on them by freely releasing code that could result in forged certificates [19].

In effect, the very existence of an exploit makes CAs who continue not to abide by CA/F rules more liable. Thus, this rather costly collision compute worked to an extent as an agent of enforcement, "correcting" (that is, enforcing a perspective upon) CAs in ways existing standards bodies were unable to do.

4.2.2 Browser developers. Alongside the issue of enforcing proper security practices on a decentralized system of certificate authorities, a separate ecosystem of browser developers exercises independent authority to accept, or reject, certificates issued by CAs.

While browser production is also decentralized, it is less so than CAs (Figure 2) [5].

According to these statistics, the majority of browsers on the web had already agreed to stop accepting SHA1 SSL certificates, even before this compute took place [8, 11]. So, regardless of what certificate authorities do, users of these browsers would have been protected from any vulnerabilities in SHA1, and the CAs would have faced additional market pressure to move away from SHA1.

If the performance of the collision compute was not necessary to change behaviors among browser developers (and thus to protect users), why was it performed? One explanation may come from the press room. Browser developers such as Mozilla and Google have received criticism for their decision to reject SHA1 certificates, even from other industry leaders such as Facebook [17], given the still-theoretical nature of the hash's vulnerability Thus, another dimension of this compute's polemic aims relates to browser PR, undercutting claims that the decision to deprecate SHA1 was premature. Crucially, browsers has a vested interest in security: browsers need their users to feel secure, as customers will flee if they do not feel safe shopping and communicating on the Internet.

5 THE POLEMICS OF ACTUALLY DOING

The prior section gave sociotechnical context for the performance of the SHA1 collision compute, giving many explanations across a wide variety of contexts. However, as of now, we lack a theory for systematically typifying these disparate explanations. In this section, I propose a definition of *polemic computation* to describe motivations for performing computes such as those above (Section 5.1). Namely, we propose that some computation is performed because there is a polemic power to doing so, and that the material resources expended on such a computation take agency in particular sociotechnical debates. We tie this theory to that of charismatic technology (Section 5.2) and to critical design (Section 5.3) in centering the material nature of performed computation in describing its agential power in sociotechnical discourses.

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 [6]. Their performance or enactment also requires specialized technical expertise in the form of labor. Polemic computes are at once feats and artifacts, which act [1] in sociotechnical debates. The following sections relate this theory of polemic computations to other theories of charismatic technology and critical design, highlighting the relevant differences to our theories.

5.2 Charismatic technology

Polemic computation can be said to "work" in part because it is animated by ideological frameworks. In the case of the SHA1 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.

In this way, polemic computation draws strongly from Ames' theory of *charismatic technology* [1]. Drawing on actor-network theory, charismatic technology would ascribe the very artifact of the computation (a material artifact produced by material means [3, 6**?**]) agency in the technosocial discourses around privacy and security. 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 debates.

As with charisma, power is central to polemic computing. Here, power plays in through the resources required to perform the compute. 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 [16]. 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 expenditure to work as an agent in technosocial discourse. Indeed, the SHA1 collision demonstrated an attack feasible only for highly resourceful actors (for now). Such actors might be a government or, apparently, Google. Thus, this collision demonstrated not only the considerable resources required to exploit SHA1, but the vast resources that Google must have, if it is able to spend so heavily on a project with essentially polemic aims.

5.3 Critical design

Another strand of research that explicitly centers the agency of technological artifacts is critical design [7]. 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 SHA1 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 [3] 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 could they could have computed instead?).

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. This section aims to outline potential avenues for future work by reviewing various classes of compute projects, highlighting ways in which they could be considered polemic, and surfacing the field of limits, 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, ecological 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 [2]. 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 facilities. Network transmissions, and the generally lower power-efficiency of commodity devices. incur additional costs in energy. These projects reduce capital overhead for those running the compute, but may exacerbate 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 useful data [2]. 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 commodity 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 examine the motivations for architectural decisions in web applications through the polemics around Web 2.0 [15], 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 [9]), the production of such an algorithm is often incidental to the production of the benchmark. Benchmarks serve to mark or legitimize 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 benchmarks [16].

7 CONCLUSION

Using the example of a particular large-scale compute, this paper highlights broader tensions about when and when not to compute. In the case of polemic computation, the social rewards are always unclear. Future work should consider broadly what we can we do to hedge our risks, not just in time and capital, but in ecological cost. We will only have more things to compute, and more things to compute them with, but how to select which to expend our increasingly precious resources on, how to use restraint?

8 ACKNOWLEDGEMENTS

Thanks to Morgan Aimes, Nick Doty, Anette Greiner, Sebastian Benthall and John Chuang for their comments and conversations.

REFERENCES

[1] Morgan G. Ames. 2015. Charismatic Technology. *Proceedings of the 5th Decennial AARHUS Conference* (2015), 109–120. DOI:http://dx.doi.org/10.1080/19447014508661941
[2] David P Anderson. 2004. Public Computing : Reconnecting People to Science. *Knowledge Creation Diffusion Utilization* (2004), 1–6. http://boinc.berkeley.edu/boinc2.pdf
[3] Jane Bennett. 2013. *Vibrant Matter: a political ecology of things*. Vol. 53. 1689–1699 pages. DOI:http://dx.doi.org/10.1017/CBO9781107415324.004 arXiv:arXiv:1011.1669v3
[4] 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
[5] Craig Buckler. 2016. Browser Trends January 2016: 12 Month Review. (2016). https://www.sitepoint.com/browser-trends-january-2016-12-month-review/
[6] Paul Dourish and Melissa Mazmanian. 2011. Media as Material: Information Representations as Material Foundations for Organizational Practice. *Proc. Int. Symp on Process Organization Studies* (2011), 1–24. DOI:http://dx.doi.org/10.1093/acprof:oso/9780199671533.003.0005
[7] Anthony Dunne and Fiona Raby. 2001. *Design Noir: The Secret Life of Electronic Objects*. Vol. 1. 176 pages. DOI:http://dx.doi.org/10.1007/s13398-014-0173-7.2 arXiv:arXiv:gr-qc/9809069v1
[8] Ryan et al Sleevi. 2014. Intent to Deprecate: SHA-1 certificates. (2014). https://groups.google.com/a/chromium.org/forum/
[9] Neal Jean, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon. 2016. Combining satellite imagery and machine learning to predict poverty. *Science (New York, N.Y.)* 353, 6301 (2016), 790–4. DOI:http://dx.doi.org/10.1126/science.aaf7894
[10] Antti Koivisto. 2017. Bug 168774 - Add a test verifying cache deduplication is not sensitive to SHA1 collision attack. (2017). https://bugs.webkit.org/show
[11] Mozilla. 2017. CA:Problematic Practices. (2017). https://wiki.mozilla.org/CA:Problematic
[12] Greg Pautsch, Duncan Roweth, and Scott Schroeder. 2016. *The Cray® XCTM Supercomputer Series: Energy-Efficient Computing*. Technical Report. Cray, Inc. 23 pages.
[13] Q-Success. 2017. Usage of SSL certificate authorities for websites. (2017). https://w3techs.com/technologies/overview/ssl
[14] Bruce Schneier. 2005. Cryptanalysis of SHA-1. (2005). https://www.schneier.com/blog/archives/2005/02/cryptanalysis
[15] 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
[16] M. Six Silberman. 2015. Information systems for the age of consequences. *First Monday* 20, 8 (2015), 1–1.
[17] Ale Stamos. 2015. The SHA-1 Sunset. (2015).
[18] 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

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

[20] 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

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

[22] 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