5-1-2013

How to Kill Copyright: A Brute-Force Approach to Content Creation

Kirk Sigmon
Cornell Law School, J.D. 2013, kas468@cornell.edu

HOW TO KILL COPYRIGHT: A BRUTE-FORCE APPROACH TO CONTENT CREATION

Kirk Sigmon[±]

I
INTRODUCTION: THE MEANING OF "NEVER"

What does "never" mean? This question was illustrated by a problem in Charles Kittel and Herbert Kroemer's textbook *Thermal Physics,* in which the authors discussed a popular hypothetical: the so-called infinite monkey theorem.[1] The authors posed a problem:

[±] Dual B.A., Wake Forest University, 2010; J.D. Candidate, Cornell Law School, 2013. The original idea for this paper comes from my friend and classmate Ryan Delaney, who first conceptualized the idea of a computer generating copyrighted material. I also want to thank other members of Prof. Oskar Liivak's Topics in Intellectual Property seminar (as well as Prof. Liivak himself), all of who provided critical ideas and feedback as this paper was written. Credit for part of the title, as well as a very preliminary version of this idea, should also be given to Robert Rogoyski. *See* Robert Rogoyski, *The Melody Machine: How to Kill Copyright, and Other Problems with Protecting Discrete Musical Elements,* 88 J. PAT. & TRADEMARK OFF. SOC'Y 347 (2006).

[1] CHARLES KITTEL & HERBERT KROEMER, THERMAL PHYSICS 53 (W H Freeman & Co 1980).

> Suppose that 10^{10} monkeys have been seated at typewriters through the age of the universe, 10^{18}. This number of monkeys is about three times greater than the present human population of the earth. We suppose that a monkey can hit 10 typewriter keys per second. A typewriter may have 44 keys; we accept lowercase in place of capital letters. Assuming that Shakespeare's *Hamlet* has 10^5 characters, will the monkeys hit upon *Hamlet?*[2]

As one may imagine from the context of the question, the point of Kittel and Kroemer's question was to illustrate that monkeys would effectively *never* type out *Hamlet*:

> The probability that a *monkey-Hamlet* will be typed in the age of the universe is approximately 10^{164316}. The probability of *Hamlet* is therefore zero in any operational sense of an event, so that the original statement . . . is nonsense: one book, much less a library, will never occur in the total literary production of the monkeys.[3]

Perhaps thankfully, this paper does not argue that monkeys could potentially produce *Hamlet*. But it does seek to prove a related point: modern technology has made it theoretically possible for a computer system (rather than a monkey) to intentionally generate copyrightable work, and copyright law may have difficulty reacting to such an innovation. In fact, this method of content generation – which I call brute-force content creation – could be a very troublesome loophole in copyright law.

In Part II, I analyze the cryptanalytic method known as brute-forcing[4] and how it could be used to generate copyrightable content such as images[5] and audio.[6] Brute-forcing, in layman's terms, is a way in which a computer generates every possible variation (or *permutation*) of a string.[7] While brute-forcing has been traditionally used to guess encrypted passwords, I explain that it could be used to generate copyrightable content.[8] Though I conclude that it is not yet

---

[2] *Id.*
[3] *Id.*
[4] *See infra* Part II.
[5] *See infra* Part II.A.
[6] *See infra* Part II.B.
[7] *See infra* Part II.
[8] *See infra* Part II.C.

technically feasible to brute-force copyrighted content,[9] I devote a short subpart to discussing how one might optimize an algorithm to make the process technically feasible.[10]

In Part III, I discuss whether or not such content could be copyrighted.[11] I first analyze the underlying doctrine that makes brute-force content creation so appealing and so possible: the doctrine of independent creation.[12] I then turn to whether or not such content would *actually* be copyrightable, focusing on how the relatively weak "modicum of creativity" standard may be a bar where an algorithm blindly creates content.[13]

In Part IV, I turn to the legal ramifications of brute-forcing copyrighted content.[14] First, I address the infringement ramifications of operating a brute-forcing algorithm intended to create every possible permutation of creative content.[15] Second, I discuss problems of contributory liability in two contexts: allowing parties to search through brute-forced content, and allowing third parties to buy and/or acquire brute-forced content.[16] Third, I discuss how brute-force content creation could potentially infringe trademark.[17] Finally, I discuss the ramifications of a blatant attack on copyright from the perspective of a legal realist.[18]

I conclude in Part V, explaining why it is a good thing that it is currently impossible to brute-force content.[19]

---

[9] *Id.*
[10] *See infra* Part II.D.
[11] *See infra* Part III.
[12] *See infra* Part III.A.
[13] *See infra* Part III.B.
[14] *See infra* Part IV.
[15] *See infra* Part IV.A.
[16] *See infra* Part IV.B.
[17] *See infra* Part IV.C.
[18] *See infra* Part IV.D.
[19] *See infra* Part V.

II
BRUTE FORCE ATTACKS 101

Before I begin an analysis of the legal ramifications of a brute-force attack on copyrightable material, I must explain how such brute-force attacks would be feasible at all. As relatively laborious as this explanation is, it helps illustrate why such an algorithm is, in consideration of modern technology, purely theoretical.

How hard is it to guess a password? The answer is, "it depends." When passwords are stored in plain text – that is, when they are stored exactly as the user enters them – all it takes for a nefarious party to acquire a user's password is access to the computer storing the user's password itself. But this is a rare event: most modern websites and services use cryptographic functions like MD5,[20] which are one-way non-reversible encryption algorithms.[21] These encryption algorithms take an input (usually a password) and generate a long, unique,[22] and un-decipherable fingerprint-like string (a "message digest").[23] The upshot of these one-way non-reversible encryption algorithms is that, even if a nefarious hacker got access to many of these "fingerprints," there would be no way for them to un-encrypt the "fingerprints" themselves.[24]

But one-way encryption algorithms like MD5 are not fool-proof: there are many ways in which someone with an one-way-encrypted string could eventually determine what a user's password is. "Rainbow tables," or pre-calculated lists of what-password-equals-what-

---

[20] MD5 is an acronym for "Message Digest Algorithm 5." *See* R. Rivest, Network Working Group, *RFC 1321: The MD5 Message-Digest Algorithm,* INTERNET ENGINEERING TASK FORCE DATATRACKER (Apr. 1992), *available at* http://datatracker.ietf.org/doc/rfc1321/.

[21] *See generally id.*

[22] This is no longer the case with MD5, which is susceptible to collision attacks. *See* Tao Xie & Dengguo Feng, *How to Find Weak Input Differences for MD5 Collision Attacks,* CRYPTOLOGY (May 30, 2009), *available at* http://eprint.iacr.org/2009/223.pdf (finding weak input differences in the MD5 protocol).

[23] R. Rivest, Network Working Group, *RFC 1321: The MD5 Message-Digest Algorithm,* INTERNET ENGINEERING TASK FORCE DATATRACKER (Apr. 1992), *available at* http://datatracker.ietf.org/doc/rfc1321/ ("It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest.").

[24] At least insofar as the mechanisms behind algorithms like MD5 remain a secret.

fingerprint, are commonplace online and relatively easy to find.[25] But these tables are often limited to what is computationally feasible: many only cover passwords up to 10 characters in length.[26] Some forms of social engineering,[27] such as pretending to be tech support and verifying a user-provided password against a "fingerprint," also work in certain circumstances.[28]

But there is another option that is relevant not only to password cracking, but also to the world of copyright: brute-force attacks.

A brute-force attack is a cryptanalytic[29] attack that exhaustively guesses every possible permutation of a string until it finds the correct solution.[30] As the title implies, brute-forcing is more or less the cryptanalytic equivalent of jamming random thin objects into a lock until it opens – it entails guessing every single possible option until one option eventually works. In the context of the cryptographic "fingerprints" mentioned above, it involves encrypting every single possible string until the right "fingerprint" comes out, which means you have input the user's password.

The number of potential variations of a string (like a password) is calculated using two variables: the number of characters usable in the target string (the "character set"), and the maximum length of the target string: [31]

$$Number\ of\ Permutations\ =\ (Character\ Set)^{(Length\ of\ String)}$$

---

[25] For example, the RainbowCrack project lists a large number of LM, NTLM, MD5, and SHA1 rainbow tables. *List of Rainbow Tables,* RAINBOWCRACK PROJECT, *available at* http://project-rainbowcrack.com/table.htm (last visited Mar. 23, 2013).

[26] *See id.* (the longest table, "md5_loweralpha-numeric#1-10," which contains passwords up to 10 characters long, only covers lower alphanumeric characters).

[27] Social engineering is the art of tricking people into divulging secret information, including passwords. This can include everything from pretending to be corporate tech support to creating false log-in forms that record a user's password. *See generally* JOHNNY LONG & KEVIN D MITNICK, NO TECH HACKING: A GUIDE TO SOCIAL ENGINEERING, DUMPSTER DIVING, AND SHOULDER SURFING (2008).

[28] *See id.*

[29] Cryptanalysis is "is the science and sometimes art of breaking cryptosystems." CHRISTOF PAAR & JAN PELZL, UNDERSTANDING CRYPTOGRAPHY: A TEXTBOOK FOR STUDENTS AND PRACTITIONERS 3 (2010).

[30] *Id.* at 7.

[31] *See* NICK MOLDOVYAN & ALEX MOLDOVYAN, INNOVATIVE CRYPTOGRAPHY 63 (2nd ed. 2007).

As the length of a string exponentially increases the number of its permutations, brute-force attacks are ridiculously inefficient against relatively lengthy passwords. By way of example, assuming a password potentially comprised of the full 95 printable ASCII characters[32] with a maximum length of 8 characters (such as the string "C()rnell"):

$$Number\ of\ Permutations\ (\text{“C()rnell”}) = 95^8 = 6{,}634{,}204{,}312{,}890{,}625$$

In other words, this password would have well over six *quadrillion* possible variations. While a computer attempting to brute force this password may not have to calculate every single one of these permutations to discover the password is "C()rnell" (it might begin with "C" before it goes to "D" and thus try "C()rnell" before it tries "D()rnell"), this large number of permutations all but guarantees that any effort to guess a password would require an incredible amount of time and computing power.[33]

In comparison, a much shorter password comprised of nothing but the 26 lower case letters of the alphabet and only 5 characters – say, the word "cases" – would only have nearly 12 million possible permutations:

$$Number\ of\ Permutations\ (\text{“cases”}) = 26^5 = 11{,}881{,}376$$

It is thus easy to understand why websites like Facebook that require a password prefer lengthy passwords using more than the lower-case alphabet:[34] doing so raises the potential number of permutations (and thus the difficulty of brute-forcing a password) exponentially.

---

[32] *See* RANDALL HYDE, THE ART OF ASSEMBLY LANGUAGE § 2.14.1 (2nd ed. 2010).

[33] For a full discussion of the computing power required to brute-force strings, see *infra* Part II.C.

[34] For example, Facebook's help guide asks that users make a password "at least 6 characters long" using "a complex combination of numbers, letters, and punctuation marks." *What is the Minimum Password Strength and How Can I Make My Password Strong?,* FACEBOOK.COM, http://www.facebook.com/help/124904560921566 (last visited Mar. 18, 2013).

A. IMAGES AND VIDEO

Because digital images can be reduced to a string in a manner not dissimilar to a password, it is theoretically feasible to brute-force an image.

Digital images are displayed using pixels – that is, miniature dots on a computer monitor that, when placed together in a matrix, display an image. For example, Google's default main page logo is 550 pixels wide and 190 pixels tall[35] – in other words, a series of 104,500 pixels. Each pixel in the Google logo is, for compression reasons, 8 bits, which allows each pixel to display one of 256 colors.[36] This is as if the Google logo was a 104,500 character string comprised of a 256 character color alphabet:

$$Number\ of\ Permutations\ (Google\ logo)\ = 256^{104{,}500}$$

Suffice to say, the number of permutations of the Google logo is rather large: too large to print here, and certainly too large to brute force. More specifically, the length of the number of permutations itself has 251,662 decimal digits – more than two hundred times more decimal digits than characters in this paragraph. This result does not even involve an image with an aesthetically pleasing number of colors: most images today use 24-bit color, which allows for 16,777,216 different possible colors in a single pixel.[37]

Even smaller, less detailed images still result in an incredible number of permutations. For example, a 1-bit image (that is, an image with only black pixels and white pixels) with 100 pixels (that is, an image 10 pixels wide and 10 pixels tall) would still have

---

[35] *See* [Google Logo], https://www.google.com/images/srpr/logo4w.png.

[36] *See id.*

[37] I am omitting a discussion of 32-bit color because it only adds 256 levels of transparency, which are unlikely to be used in most copyrighted images. *See Bit Depth,* U. OF CAMBRIDGE DEP'T OF CHEMICAL ENGINEERING AND BIOTECHNOLOGY, http://www.ceb.cam.ac.uk/pages/bit-depth.html (last visited Mar. 19, 2013). For similar reasons, I omit a discussion of all forms of so-called "deep color" consisting of more than 24 bits per pixel, as such a color gamut is unnecessary for brute-forcing purposes. *See* JOE CELKO, JOE CELKO'S THINKING IN SETS: AUXILIARY, TEMPORAL, AND VIRTUAL TABLES IN SQL 168 (Morgan Kaufmann 2008).

1,267,650,600,228,229,401,496,703,205,376 (that is, over one nonillion) permutations. That is more permutations than there are grains of sand on the earth.[38]

There is an alternative approach to brute-forcing very large images with an astronomical numbers of pixels: bitwise brute-forcing. All computer data is currently[39] stored as strings of zeroes and ones, known as bits.[40] Instead of brute-forcing an image pixel-by-pixel, it may be more feasible to brute-force larger images bit-by-bit. For example, wallpaper images often have a resolution of 1920 pixels by 1080 pixels, which is the width and height of a 1080p/1080i screen.[41] When somewhat compressed[42] into a JPEG file, such an image with a bit depth of 24 bits (or 16,777,216 colors) can be as small as 490 kilobytes, or 4,014,080 bits.[43] The difference in permutations can be staggering:

$$Number\ of\ Permutations\ (1920x1080\ image)\ [by\ pixel]\ = 16{,}777{,}216^{2{,}304{,}000}$$

$$\approx\ a\ number\ with\ over\ 16.6\ million\ decimal\ digits$$

$$Number\ of\ Permutations\ (1920x1080\ image)\ [JPEG,^{44}\ bitwise]\ = 2^{4{,}014{,}080}$$

$$\approx\ a\ number\ with\ over\ 1.2\ million\ decimal\ digits$$

---

[38] Researchers estimate that there are approximately 7.5 x 10^{18} (or roughly seven quintillion) grains of sand on the earth. *See* Robert Krulwich, *Which Is Greater, The Number Of Sand Grains On Earth Or Stars In The Sky?,* NPR KRULWICH WONDERS (Sept. 17, 2012), *available at* http://www.npr.org/blogs/krulwich/2012/09/17/161096233/which-is-greater-the-number-of-sand-grains-on-earth-or-stars-in-the-sky.

[39] Quantum computing "qbits" may replace traditional bits in the future. *See* TZVETAN S. METODI, ARVIN I. FARUQUE & FREDERIC T. CHONG, QUANTUM COMPUTING FOR COMPUTER ARCHITECTS 7 (2011) (discussing classical bits and quantum signal states).

[40] *See id.*

[41] *See generally* EUR. BROADCASTING UNION, HIGH DEFINITION (HD) IMAGE FORMATS FOR TELEVISION PRODUCTION (Dec. 2004), *available at* http://web.archive.org/web/20091229093957/http://tech.ebu.ch/docs/tech/tech3299.pdf.

[42] Using a JPEG quality rating of 70 in Adobe Photoshop CS6. This is a *very* rough approximation – JPEG images of the same resolution often vary in terms of their quality and size. A true JPEG-based brute-forcing algorithm would unquestionably need to account for such variation.

[43] Assuming "kilobyte" means 8,192 bits.

[44] This is an oversimplified example of brute-forcing JPEG. To actually conduct JPEG brute-forcing, the algorithm would have to somewhat intelligently brute-force based upon the JPEG File Interchange Format, which would include adding, among other things, file headers. *See generally* ERIC HAMILTON, JPEG FILE INTERCHANGE FORMAT VERSION 1.02 (1992), *available at* http://www.w3.org/Graphics/JPEG/jfif3.pdf.

Thus, in certain circumstances, it may be efficient to abandon the pixel-by-pixel approach described above and instead simply brute-force an image bit-by-bit.

Using similar principles, it would be entirely feasible to turn any of these images into video. Such a video could be comprised of already brute-forced images, which themselves become the character set for a string comprised of frames. For example, at 24 frames per second, a one-hour silent movie the size of the Google logo would simply add another exponent:

$$Number\ of\ Permutations\ (one\ hour\ of\ silent\ video)\ = 256^{104,500^{86,400}}$$

B. AUDIO

Like images, audio can be expressed as a series of bits. Thus, just like how images can be brute-forced bit-by-bit,[45] audio could theoretically be brute-forced bit-by-bit.

Audio files (such as MP3 files) of songs are usually available in a wide range of bit rates; however, enthusiast testing has generally come to the conclusion that few listeners can discern any difference between files with bit rates above 160kbit/sec.[46] Thankfully, when creating music, it is not necessary to generate audio of a perfect quality. Thus, the slightly distorted (but nonetheless arguably listenable) 128kbit/sec constant bit rate is sufficient for brute-forcing purposes.[47]

128kbit/sec means 131,072 bits per second, where a bit is a Boolean value of zero or one. Thus:

---

[45] *See supra* Part II.A.

[46] The "Great MP3 Bitrate Experiment" came to the conclusion that there was virtually no perceivable difference in the audio quality of 160kbit/sec VBR (variable bit rate), 320kbit/sec CBR (constant bit rate), 192kbit/sec VBR, and raw CD audio. Jeff Atwood, *Concluding the Great MP3 Bitrate Experiment,* CODING HORROR (June 27, 2012), http://www.codinghorror.com/blog/2012/06/concluding-the-great-mp3-bitrate-experiment.html.

[47] The 128kbit/sec bit rate is admittedly insufficient for high quality audio. *Id.* I nonetheless use 128kbit/sec for calculations for two reasons: first, perfect audio quality is not strictly necessary to produce copyrightable material, and second, the bit rate of the audio file directly influences the size of the exponent in determining permutations (meaning a lower bitrate begets fewer permutations). Suffice to say, the lower the bitrate, the better.

$$Number\ of\ Permutations\ (one\ second)\ [128\ kbit/sec] = 2^{131,072}$$

That is, a number with 39,457 decimal digits. As the average song is four minutes long,[48] this means that the average song has 240 times the potential bits:

$$Number\ of\ Permutations\ (four\ minutes)\ [128\ kbit/sec] = 2^{31,457,280}$$

Or, a number with over 9.4 million decimal digits.

Some audio brute-forcing could also be achieved by brute-forcing the MIDI format. The MIDI[49] data format is a simplistic way in which instruments can be sequenced and re-played digitally.[50] Because the format does not store actual recordings and rather stores a sequence of notes to re-play a song (like a sort of digital sheet music), MIDI does not act as an audio recording,[51] and thus MIDI would be a poor replacement for most songs except for certain forms of audio (such as very basic melodies and compositions).

C. FEASIBILITY

As repeatedly emphasized, brute-forcing is a computationally expensive, inefficient, and normatively over-the-top approach to generating anything. Bluntly, though it may be *theoretically* feasible to brute-force copyrighted material, it is anything but *technically* feasible to do so with current technology.

---

[48] Michael Twardos, *Probability Distribution of Song Length in a Collection of Itunes Libraries,* THE INFORMATION DIET (Nov. 16, 2011), http://theinformationdiet.blogspot.com/2011/11/probability-distribution-of-song-length.html ("The distribution shows the relative likelihood of the length of a song. This plot was calculated from over 70,000 songs from 12 (American) libraries. The median of this plot is 231 seconds ans the mean is at 242 seconds. This observation may indicate something fundamental about people(culturally or biologically): we like songs that are almost exactly 4 minutes. As you move away from the 4 minute mark, the probability drops in similar amounts (the plot is symmetric-ish).").

[49] MIDI stands for "Musical Instrument Digital Interface." *See* JEFFREY HASS, INTRODUCTION TO COMPUTER MUSIC: VOLUME ONE ch. 3, *available at* http://www.indiana.edu/~emusic/etext/MIDI/chapter3_MIDI.shtml.

[50] *See generally id.* A very similar idea was discussed in the context of a "Melody Machine" by Robert Rogoyski. Robert Rogoyski, *The Melody Machine: How to Kill Copyright, and Other Problems with Protecting Discrete Musical Elements,* 88 J. PAT. & TRADEMARK OFF. SOC'Y 347, 351 (2006).

[51] *See id.*

Scholarly work discussing the feasibility of brute-forcing rarely goes beyond simple examples because it usually needs not do so. For the most rudimentary of brute-forcing algorithms, the math is simple: the total number of permutations is divided by the number of permutations that can be processed by a computer in some time unit:[52]

$$\frac{(Character\ Set)^{(Lengthof\ String)}}{(Permutations\ per\ time\ unit)}$$

For example, Raza et al. have calculated that a single computer calculating 1,000 password permutations per second could brute-force an 8-character password comprised of lower case letters (that is, 26^8 or 208,827,064,576 permutations) in roughly 58,007.52 hours, or a little over six years.[53] As the Sun is estimated to burn out in six billion years,[54] it would burn out well before that same computer could even begin to generate all of the permutations of a four-minute song.[55]

But 1,000 password permutations per second is an absolute joke for any real time expenditure analysis, as some security websites estimate that even the ancient Pentium 100 processor could guess 10,000 passwords per second.[56] More modern technology must be used to realistically estimate the time to calculate permutations.

---

[52] *See* Raza et al., *A Survey of Password Attacks and Comparative Analysis on Methods for Secure Authentication,* 19 (4) WORLD APPLIED SCI. J., 439, 439 (2012); *see also* Jain et al., *New Modified 256-bit Message Digest Algorithm Based on Existing Algorithms,* 3:1 J. COMPUTING TECH. 2278–3814 (July 2012) (analyzing brute force time against various key sizes and permutations). Note with extremely long strings, more time is necessarily spent storing and processing those strings, implicating a different formula.

[53] Raza et al., *A Survey of Password Attacks and Comparative Analysis on Methods for Secure Authentication,* 19 (4) WORLD APPLIED SCI. J., 439, 439 (2012).

[54] *See* Fraser Cain, *Life of the Sun,* UNIVERSE TODAY (Mar. 10, 2012), http://www.universetoday.com/18847/life-of-the-sun/.

[55] The calculation is too large to print, but suffice to say, a number with 924 thousand decimal places does not become much smaller when divided by the 31,540,000 seconds there are in a year. That number of years is certainly larger than the six billion years of hydrogen in the Sun. *See id.*

[56] *See Password Recovery Speeds,* LOCKDOWN.CO.UK – THE HOME COMPUTER SECURITY CENTRE, http://www.lockdown.co.uk/?pg=combi#classA (last visited Mar. 13, 2013).

In December of 2012, password cracking expert Jeremi Gosney unveiled a Linux and OpenCL-based GPU server cluster comprising of 5 servers utilizing 25 AMD Radeon graphics cards.[57] This cluster has computing power capable of guessing 350 billion passwords per second.[58] The servers require 7kW of electricity when operational[59] and likely cost well over $10,000.[60]

But even Gosney's veritable supercomputer(s) couldn't take on creative content.

Would Gosney's servers be able to brute-force the Google logo? Not in the Sun's lifetime.[61] As explained above, the Google logo has $256^{104,500}$ possible permutations.[62] The number of years it would take to brute-force this password with a single one of Gosney's server clusters is so large that, like the number of permutations of the logo, it cannot even be printed here. Specifically, the number of years it would take to brute-force the Google logo has 251,643 decimal digits.

---

[57] Dan Goodin, *25-GPU Cluster Cracks Every Standard Windows Password in <6 Hours,* ARS TECHNICA (Dec. 9, 2012), http://arstechnica.com/security/2012/12/25-gpu-cluster-cracks-every-standard-windows-password-in-6-hours/.

[58] *Id.* This is when the GPU cluster brute-forces NTLM passwords and not more complex hashes. Arguably, the fact that such calculations are based on NTLM encryptions is insignificant: a brute-force content creation algorithm would ideally be as computationally simple as the NTLM protocol, if not more so.

[59] JEREMI M. GOSNEY, PASSWORD CRACKING HPC, PASSWORDS^12 SECURITY CONF. 17 (Dec. 13, 2012), *available at* http://passwords12.at.ifi.uio.no/Jeremi_Gosney_Password_Cracking_HPC_Passwords12.pdf.

[60] The cluster contains 10 AMD Radeon HD 7970s, 4 AMD Radeon HD 5970s, 3 AMD Radeon HD 6990s, and 1 AMD Radeon HD 5870. *Id.* The cost of a HD 7970 is approximately $479. *See* Hassan Mujtaba, *AMD Officially Announces Price-Cuts for Radeon HD 7000 Series, HD 7970 Now Available for $479 MSRP,* WCCF TECH, http://wccftech.com/amd-officially-announces-pricecuts-radeon-hd-7000-series-hd-7970-479-msrp/ (last visited Mar. 23, 2013). The price of a dual-GPU HD 5970 is approximately $599. Matthew Murray, *AMD Releases Dual-GPU Radeon HD 5970 Card,* PC MAG. (Nov. 17, 2009), *available at* http://www.pcmag.com/article2/0,2817,2356053,00.asp. The price of an HD 6990 is approximately $700. Sal Cangeloso, *AMD Announces the Ridiculously Powerful $700 Radeon HD 6990 Graphics Card,* GEEK.COM (Mar. 8, 2011), *available at* http://www.geek.com/chips/amd-announces-the-ridiculously-powerful-700-radeon-hd-6990-graphics-card-1320191/. The price of an HD 5870 is approximately $379. Ryan Smith, *AMD's Radeon HD 5870: Bringing About the Next Generation of GPUs,* ANANDTECH (Sept. 23, 2009), http://www.anandtech.com/show/2841. This means that the graphics cards in Gosney's servers could have cost as much as $9,665.00, not including the enclosure, processors, motherboards, and the like. But many of these prices (except for 7970's price) are launch prices, meaning an equivalent server could be slightly cheaper today. Thus, for simplicity's sake and without better information, I estimate the price of one of Gosney's servers to be roughly $10,000.

[61] The Sun is estimated to burn out in six billion years. *See* Fraser Cain, *Life of the Sun,* UNIVERSE TODAY (Mar. 10, 2012), http://www.universetoday.com/18847/life-of-the-sun/.

[62] *See supra* Part II.A.

Gosney's servers could not even brute-force a simple poem. For example, W.B. Yeats' *He wishes for the Cloths of Heaven* is a relatively short poem with 327 characters, including spaces.[63] Using the ASCII 95-character set,[64] this would entail 95^{327} permutations. Even with the power of Gosney's servers, it would still take more years than ever documented in human history to brute force Yeats' poem – more specifically, a number of years so large it has 628 decimal digits.

These astronomical figures are not made any more tolerable by throwing more servers at the problem. Adding another server would certainly double the workload, but would cost yet another $10,000 and yet another 7kW of electricity.[65] Assuming one had the 2011 GDP of the United States – $14.99 trillion dollars[66] – this means that one could only buy at best 1,499,000,000 of Gosney's servers, not including electrical/storage/facility/other expenses. The result? It would still require a number of years with 619 (that is, 9 fewer) decimal digits to brute-force Yeats' poem and a number of years with 251,633 (that is, 10 fewer) decimal digits to brute-force the Google logo!

These rough calculations only scratch the surface of the feasibility problem – storing permutations is also an issue. Take, again, the Google logo, an image with $256^{104,500}$ possible permutations.[67] The Google logo is approximately 18 kilobytes in size, which is approximately 20 kilobytes as stored on a hard disk.[68] To brute-force and store every possible permutation of the Google logo, a programmer would have to make available so many terabytes that the number

---

[63] *See* W. B. YEATS, THE WIND AMONG THE REEDS 26 (Kessinger Publ'g 2004).

[64] *See* RANDALL HYDE, THE ART OF ASSEMBLY LANGUAGE § 2.14.1 (2nd ed. 2010).

[65] *See* JEREMI M. GOSNEY, PASSWORD CRACKING HPC, PASSWORDS^12 SECURITY CONFERENCE 17 (Dec. 13, 2012), *available at* http://passwords12.at.ifi.uio.no/Jeremi_Gosney_Password_Cracking_HPC_Passwords12.pdf.

[66] *Data: United States,* THE WORLD BANK, http://data.worldbank.org/country/united-states (last visited Mar. 23, 2013).

[67] *See supra* Part II.A; *see also* [Google Logo], https://www.google.com/images/srpr/logo4w.png.

[68] *See id.*

of terabytes alone has over 250,000 decimal digits.[69] This is exponentially more storage than the telecommunications capacity of the entire world.[70] Yeats' poem does not fare much better – though in plain-text it is only 335 bytes (or 0.327148 kilobytes),[71] it would require a number of terabytes of storage with 851 decimal digits[72] – that is, still well over the entire telecommunications capacity of the entire world.[73]

Thus, Kroemer and Kittel were as right with computers as they were with monkeys:[74] if the entire GDP of the United States can barely influence the amount of time it would take to brute-force a copyrighted work, then it would be truly impossible to do so with current technology absent some sort of ground-breaking invention,[75] an exponential increase in worldwide computing power,[76] or dumb luck.[77]

D. "SMARTER" BRUTE-FORCE ATTACKS

The above calculations illustrate quite well the inefficiency of brute-force attacks – as complexity raises the possible number of permutations exponentially, brute-force attacks are all but useless except for guessing the shortest of strings. Perhaps thankfully, beyond leveraging the

---

[69] Where 1024 bytes = 1 kilobyte, 1024 kilobytes = 1 megabyte, 1024 megabytes = 1 gigabyte, and 1024 gigabytes = 1 terabyte. ITL EDUC. SOLUTIONS LTD., INTRODUCTION TO COMPUTER SCIENCE 109 (2nd ed. 2011).

[70] The entire effective capacity in 2007 was calculated to be 65,000 petabytes. *See* Martin Hilbert & Priscilla López, *The World's Technological Capacity to Store, Communicate, and Compute Information,* 223:6205 SCI. 60 (Apr. 2011).

[71] *See* W. B. YEATS, THE WIND AMONG THE REEDS 26 (Kessinger Publ'g 2004).

[72] Where 1024 bytes = 1 kilobyte, 1024 kilobytes = 1 megabyte, 1024 megabytes = 1 gigabyte, and 1024 gigabytes = 1 terabyte. ITL EDUC. SOLUTIONS LTD., INTRODUCTION TO COMPUTER SCIENCE 109 (2nd ed. 2011).

[73] *See* Martin Hilbert & Priscilla López, *The World's Technological Capacity to Store, Communicate, and Compute Information,* 223:6205 SCI. 60 (Apr. 2011).

[74] *See supra* Part I.

[75] Such as the fascinating technology of quantum computing. *See generally* TZVETAN S. METODI, ARVIN I. FARUQUE & FREDERIC T. CHONG, QUANTUM COMPUTING FOR COMPUTER ARCHITECTS (2011).

[76] The numbers discussed in this subpart indicate that it would have to be a very large exponential increase, far beyond the scope of Moore's Law (which may be an increasingly poor benchmark). *See* S. Borkar, *Obeying Moore's Law Beyond 0.18 Micron [Microprocessor Design],* PROCEEDINGS OF THE 13TH ANNUAL IEEE INT'L ASIC/SOC CONF. (2001).

[77] Matt Kane, a Chicago artist, purported to make this possible through a website called "PixelMonkeys," which outputs a single random permutation of an image based upon input parameters. Kane thus purported to leave the possibility that a copyrighted work would be duplicated up to chance. *See* Matt Kane, *The Pixel Monkeys Theory,* PIXELMONKEYS.COM, http://www.pixelmonkeys.org/#theory (last visited Mar. 23, 2013).

entire GDP of the United States,[78] there is hope for those looking to brute-force the content industry: "smart" methods of brute-forcing. As will be explained later,[79] "smart" brute-forcing solves two problems: not only can "smart" brute-forcing exponentially reduce the number of permutations for any given string, but it can also solve various issues related to making those permutations copyrightable.[80]

Any attempt to make brute-forcing "smarter" must fulfill a very important requirement: the processing time added from the addition of "smart" code in the algorithm must be less than the processing time expended by generating the unnecessary permutations. There is no point to adding "smart" code to an algorithm to reduce the possible number of permutations when there is no processing time saved (or, worse yet, where there is processing time added) by doing so. Because such balancing would necessarily occur on a case-by-case basis, this subpart can unfortunately only discuss the topic obliquely.

One way to exponentially reduce the number of permutations for any given format is to brute-force with large chunks of data, as opposed to individual bits or characters. For example, to brute-force a novel, it would not be necessary or efficient to guess every single letter of that novel – rather, one could save time by brute-forcing using a character set composed of dictionary words. While this increases the character set, it also lessens the length of the string, and thus the resulting number of permutations is shortened exponentially.

The efficiency of this "chunk-based" brute-forcing is best illustrated with poetry. Robert Frost's poem *The Road Not Taken* is 729 characters (including spaces), but only 144 words.[81]

---

[78] *See supra* Part II.C.
[79] *See infra* Part III.B.
[80] *See id.*
[81] *See* ROBERT FROST, THE ROAD NOT TAKEN, BIRCHES, AND OTHER POEMS 9 (Coyote Canyon Press 2010).

Ignoring punctuation and formatting, the difference in possible permutations is quite evident, even taking into account the approximately 250,000 different words in the English dictionary:[82]

$$Permutations\ (Per\ character)\ = 95^{729}\ \approx\ A\ number\ with\ 1{,}442\ decimal\ digits$$

$$Permutations\ (Per\ word)\ = 250{,}000^{144}\ \approx\ A\ number\ with\ 778\ decimal\ digits$$

Thus, when the length of the possible string being brute-forced is lessened by grouping the character set into "chunks," the number of permutations is lessened greatly. These "chunks" could, at least theoretically, be anything – short sounds, small images, digital simulations of paintbrushes, etc.

Brute-forcing can also be made "smarter" by the use of creative rules and an algorithm that "learns." Take, again, a poem. Suffice to say, few poets would write a poem that repeats the same word incessantly, meaning that an algorithm could skip over a possible permutation that involves the same word repeated more than three times in a row. Similarly, the same algorithm could learn basic linguistic rules, such as the operation of adjectives and adverbs and the use of articles such as "a" and "an." A truly gifted programmer could also construct an algorithm that mimics common linguistic tropes such as rhyming, alliteration, and the like to further limit the number of potential permutations. And it goes without saying that the programmer able to create a creatively gifted artificial neural network[83] would not only make their brute-forcing program smarter, but would also revolutionize the entire computing world.

---

[82] *How Many Words Are There in the English Language?,* OXFORD DICTIONARIES, http://oxforddictionaries.com/us/words/how-many-words-are-there-in-the-english-language (last visited Mar. 23, 2013) ("[T]here are, at the very least, a quarter of a million distinct English words, excluding inflections, and words from technical and regional vocabulary not covered by the OED, or words not yet added to the published dictionary, of which perhaps 20 per cent are no longer in current use.").

[83] Artificial neural networks are "nonlinear mapping systems whose structure is loosely based on principles observed in the nervous system of humans and animals." RUSSELL D. REED & ROBERT J. MARKS, NEURAL SMITHING : SUPERVISED LEARNING IN FEEDFORWARD ARTIFICIAL NEURAL NETWORKS 1 (Bradford Books 1999).

III
CAN BRUTE-FORCED WORKS BE COPYRIGHTED?

Despite the fact that the preceding Part II all but stated that brute-forcing media is technically impossible,[84] there is still a remote possibility that a clever programmer (or, more realistically, a collection of clever programmers) could brute-force copyright. Though it would be nearly impossible to brute-force a four-minute song using the most inefficient brute-forcing methods,[85] using tactics similar to the "smart" brute-forcing tactics describe above,[86] it is at least conceivable that one could construct a music-making algorithm that generated Top 40 songs at some point in the future (and hopefully before the Sun burns out[87]).

But technical feasibility is only the beginning. Assuming, for the sake of the argument, that the brute-forcing of any form of media was *technically* possible, could what an algorithm generates actually be copyrighted? That rather important question is precisely the subject this Part addresses.

A. INDEPENDENT CREATION BY AN ALGORITHM

The doctrine of independent creation is the crux upon which the entire concept of brute-forcing copyright rests.

Copyright law, as its name entails, prohibits unauthorized copying.[88] To establish a case of copyright infringement *vis-à-vis* violation of the right to reproduce a work,[89] a plaintiff must

---

[84] *See supra* Part II.
[85] *See supra* Part II.B.
[86] *See supra* Part II.D.
[87] *See supra* Part II.C (briefly discussing the lifespan of the Sun as a reference against the feasibility of brute-force attacks).
[88] ROBERT P. MERGES, PETER SETH MENELL & MARK A. LEMLEY, INTELLECTUAL PROPERTY IN THE NEW TECHNOLOGICAL AGE 417 (6th ed. Aspen Law & Bus. 2012) ("Copyrights do not give their owner the exclusive right to prevent others from making, using, or selling their creations. Rather, they give the author only the right to prevent *unauthorized copying* of their works, as well as the right to prevent some limited types of uses of those works").
[89] *See* 17 U.S.C. § 106(1).

prove either directly or circumstantially that the defendant copied her copyrighted work.[90] Because direct copying is often difficult to prove, plaintiffs can circumstantially prove copying by showing that a defendant had (1) access to the plaintiff's work and that there was (2) a substantial similarity between the defendant's work and the plaintiff's work.[91]

The doctrine of independent creation, a "cornerstone" of copyright law,[92] is a rebuttal of a plaintiff's case of direct or circumstantial copyright infringement which argues that the alleged infringer created the allegedly infringing work wholly independently of the copyright holder's work and usually without knowledge of that work.[93] In this way, the doctrine tracks the requirement that *all* creative works must be independently created to be original and therefore amenable to copyrighting.[94]

In *Mazer v. Stein,*[95] the Supreme Court provided an example of when the doctrine of independent creation applies:

> Two men, each a perfectionist, independently make maps of the same territory. Though the maps are identical each may obtain the exclusive right to make copies of his particular map, and yet neither will infringe the other's copyright.[96]

But, of course, independent creation does not only protect creative works that are based off of some constant referent, like how maps are (hopefully) based upon the geography of a region.[97] As eloquently stated by Learned Hand in *Sheldon v. Metro-Goldwyn Pictures Corp.,*[98]

---

[90] MERGES, MENELL & LEMLEY, *supra* note 88, at 520; *see also* Arnstein v. Porter, 154 F.2d 464 (2d Cir. 1946); Ty, Inc. v. GMA Accessories, Inc., 132 F.3d 1167 (7th Cir. 1997). This test is also phrased in a way that requires proof of "improper appropriation," *see* MERGES, MENELL & LEMLEY, *supra* note 88, at 527; however, this is not discussed here because brute-force content creation does not by definition involve the sort of character/trope/detail specific variations typically involved in such an analysis. *See* Nichols v. Universal Pictures Corp., 45 F.2d 119 (2d Cir. 1930) (comparing two different plays by analyzing such details).

[91] Arnstein v. Porter, 154 F.2d 464 (2d Cir. 1946); Ty, Inc. v. GMA Accessories, Inc., 132 F.3d 1167 (7th Cir. 1997); *see also* MERGES, MENELL & LEMLEY, *supra* note 88, at 520–527.

[92] 3 WILLIAM F. PATRY, PATRY ON COPYRIGHT § 9:32 (2012).

[93] *See id; see also* Calhoun v. Lillenas Publ'g, 298 F.3d 1228, 1232–33 (11th Cir. 2002).

[94] MERGES, MENELL & LEMLEY, *supra* note 88, at 421.

[95] 347 U.S. 201, 74 S. Ct. 460, 98 L. Ed. 630 (1954).

[96] Mazer v. Stein, 347 U.S. 201, 217–18, 74 S. Ct. 460, 470–71, 98 L. Ed. 630 (1954) (*citing* Fred Fisher, Inc., v. Dillingham, 298 F. 145, 151 (S.D.N.Y. 1924)).

> [I]f by some magic a man who had never known it were to compose anew Keats's *Ode on a Grecian Urn,* he would be an "author," and, if he copyrighted it, others might not copy that poem, though they might of course copy Keats's.[99]

In other words, when two parties create the same work, both independently own a copyright in their respective works even if the two works are exactly the same.[100] This is not the case in patent law, which gives a patent holder the right to prevent the independent creation of a patented invention for the duration of the patent.[101]

Because an argument of independent creation often implicitly concedes the second prong of a circumstantial copying case[102] – that is, the similarity of the plaintiff's work and the defendant's work – independent creation cases often hinge on the degree of access that the defendant had to the plaintiff's work.[103] Along these lines, the best invocations of the independent creation doctrine are made by defendants who could not have possibly had any access to the plaintiff's work – for example, a musician on a remote island with no radio

---

[97] Naturally, in the context of maps and other materials based upon constant referents, courts have found the doctrine of independent creation quite useful. *See, e.g.,* Fred Fisher, Inc., v. Dillingham, 298 F. 145, 150–51 (S.D.N.Y. 1924) ("No one doubts that two directories, independently made, are each entitled to copyright, regardless of their similarity, even though it amount to identity. Each being the result of original work, the second will be protected, quite regardless of its lack of novelty. But the best instance is in the case of maps. Here, if each be faithful, identity is inevitable, because each seeks only to set down the same facts in precisely the same relations to each other. So far as each is successful, each will be exactly the same.").

[98] 81 F.2d 49 (2d Cir. 1936).

[99] Sheldon v. Metro-Goldwyn Pictures Corp., 81 F.2d 49, 54 (2d Cir. 1936).

[100] *See* Fred Fisher, Inc. v. Dillingham, 298 F. 145, 150–51 (S.D.N.Y. 1924).

[101] *See* 2 JOHN MILLS, DONALD REILEY & ROBERT HIGHLEY, PAT. L. FUNDAMENTALS § 6:3 (2d ed.) ("A copyright, unlike a patent, does not give its owner 'the right to exclude' anyone who created the work independently of the author through whom the copyright is derived."); Clarisa Long, *Information Costs in Patent and Copyright,* 90 VA. L. REV. 465, 525 (2004) ("Independent creation is no defense to a claim of patent infringement.").

[102] With alternate pleading and the like, this is not always the case.

[103] Fred Fisher, Inc., v. Dillingham, 298 F. 145, 150–51 (S.D.N.Y. 1924) (discussing how similarity is inherent in the independent creation of "faithful" maps). Note that some courts view the relationship between similarity and access as a sliding scale. *See, e.g.,* Ty, Inc. v. GMA Accessories, Inc., 132 F.3d 1167, 1170 (7th Cir. 1997) ("If, therefore, two works are so similar as to make it highly probable that the later one is a copy of the earlier one, the issue of access need not be addressed separately, since if the later work was a copy its creator must have had access to the original.") (*citing* Selle v. Gibb, 741 F.2d 896, 901 (7th Cir. 1984)); Carew v. R.K.O. Radio Pictures, 43 F. Supp. 199 (S.D. Cal. 1942) ("If there is identity, then access is, in itself, of no importance whatsoever"). Ostensibly, this sliding scale is merely a presumption that can be rebutted by showing the legitimate possibility of independent creation.

access.[104] Of course, a defendant does not need to abscond to a remote island to claim independent creation: even where plausible arguments are made that a defendant *may* have had access to a work, courts are reluctant to find access without a strong showing of substantial similarity, and even then such a finding is not automatic.[105]

It almost goes without saying that a brute-force content creation algorithm is the ideal artificial musician on a remote island without radio access.[106] A brute-force content creation algorithm cannot deliberately or accidentally duplicate copyrighted works unless programmed to do so. When such an algorithm creates works substantially similar to the others' works, it does so without even the remotest hint of access, providing the perfect defense to any allegation of copying.

But this application of independent creation to brute-force content creation may rely upon an unnecessarily formal understanding of independent creation. Professor Clarisa Long of the University of Virginia has argued that the independent creation privilege exists, at least in part, as a mechanism to impose an actual notice requirement on alleged infringers.[107] This makes a lot of sense: the common examples of independent creation are not where an infringer intentionally avoids copyrighted material to "accidentally" duplicate it, but rather where that infringer is unintentionally unaware of the existence of similar copyrighted material.[108] If Long is correct and the doctrine of independent creation is a question of actual notice, then a brute-force content

---

[104] 3 WILLIAM F. PATRY, PATRY ON COPYRIGHT § 9:36 (2012).

[105] *See, e.g.,* Sarkadi v. Wiman, 135 F.2d 1002 (2d Cir. 1943) (finding plaintiff's showings of access and indirect showings of access by similarity insufficient); *see also* 2 WILLIAM F. PATRY, PATRY ON COPYRIGHT § 3:28 (2012) ("Independent creation may, nevertheless, still be found where plaintiff referred to (but did not copy) another's work, and, where plaintiff received only ideas or suggestions from others.").

[106] *See* 3 WILLIAM F. PATRY, PATRY ON COPYRIGHT § 9:36 (2012) (giving the remote island hypothetical).

[107] Clarisa Long, *Information Costs in Patent and Copyright,* 90 VA. L. REV. 465, 529 (2004); *see also* William M. Landes & Richard A. Posner, *An Economic Analysis of Copyright Law,* 18 J. LEGAL STUD. 325, 345–46 (1989) (discussing search costs avoided through the doctrine of independent creation).

[108] *See supra* Part II.A (providing two examples from case law, both involving infringers that did not intentionally avoid other works).

creation algorithm may not independently create content. This is because the creation of a brute-force content creation algorithm designed and intended to independently create arguably involves constructive notice of the existence of potential infringement, as the operator of such an algorithm would have reason to know the algorithm could easily infringe existing copyrights.

B. "CREATION" BY FORCE

Though a brute-force content creation algorithm is almost perfectly amenable to the doctrine of independent creation, that does not necessarily mean what it creates is copyrightable. In fact, there are reasons why what it generates is likely *not* copyrightable. This has interesting ramifications for a hypothetical brute-forcing business entity: while a valid copyright is not a prerequisite to invoking the doctrine of independent creation, a valid copyright *is* necessary to market generated permutations without inviting competitors to copy the permutations as they wish.[109]

Copyright protection exists in "original works of authorship" that are "fixed in any tangible medium of expression."[110] "Original[ity]," as developed by the courts, entails (1) independent creation (discussed above[111]) and a (2) "modicum of creativity."[112] The bar for this modicum of creativity requirement is incredibly low: an "author" must merely contribute something more than a "merely trivial" variation,[113] and courts explicitly refuse to judge the artistic merit of a work.[114]

---

[109] Though, as I discuss below, an entity could craft a *ProCD*-esque contract to bind users of the permutations to limit their dissemination or use of the permutations.

[110] 17 U.S.C. § 102.

[111] *See supra* Part III.A. The defensive use of independent creation involves the same inquiry as the use of independent creation involved in establishing that some creative work can be copyrighted. 2 WILLIAM F. PATRY, PATRY ON COPYRIGHT § 3:30 (2012) (discussing both the requirement for copyright and the defense as substantially the same).

[112] MERGES, MENELL & LEMLEY, *supra* note 88, at 421.

[113] Alfred Bell & Co. v. Catalda Fine Arts, Inc., 191 F.2d 99, 102–103 (2d Cir. 1951).

[114] Bleistein v. Donaldson Lithographing Co., 188 U.S. 239, 251–52 (1903).

Though it is almost certain given the discussion above that a brute-force content creation algorithm independently creates,[115] it is questionable whether or not an algorithm can possess a "modicum of creativity." Admittedly, as stated above, courts set the bar of creativity for a copyrightable work at the floor in analyzing copyrightable material.[116] In fact, given the weakness of the modicum of creativity requirement, older courts characterized the originality requirement as merely a re-statement of the prohibition on copying.[117] But recent Supreme Court precedent indicates that the creativity requirement will be enforced where, for example, an alleged "creation" is merely an alphabetic arrangement of names in a directory with no creativity involved in the arrangement of those names.[118]

It is not entirely clear that this strengthened modicum of creativity requirement is met where a computer is programmed to blindly generate every possible permutation of a type of creative work. The generation of permutations does not necessarily entail plausibly creative activity, such as making "choices as to [the] selection and arrangement" of data.[119] If a mechanically and functionally arranged "list" of names in alphabetical order is not copyrightable,[120] a court may refuse to find that a "list" of every possible permutation of a format of creative work, similarly mechanically and functionally arranged, entails a modicum of creativity.

---

[115] *See supra* Part III.A.

[116] *See* Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., Inc., 499 U.S. 340, 345, 111 S. Ct. 1282, 1287, 113 L. Ed. 2d 358 (1991) ("To be sure, the requisite level of creativity is extremely low; even a slight amount will suffice. The vast majority of works make the grade quite easily, as they possess some creative spark, 'no matter how crude, humble or obvious' it might be.").

[117] *See, e.g.,* Alfred Bell & Co. v. Catalda Fine Arts, Inc., 191 F.2d 99, 102–103 (2d Cir. 1951).

[118] *See* Feist Publ'ns, Inc. v. Rural Tel. Serv. Co., Inc., 499 U.S. 340, 111 S. Ct. 1282, 113 L. Ed. 2d 358 (1991). This ruling allegedly reinstated prior law which prohibited so-called "sweat of the brow" copyrights – that is, copyrights for works that resulted from mere labor, not creativity. *See* 2 WILLIAM F. PATRY, PATRY ON COPYRIGHT § 3:27 (2012).

[119] *Id.* at 348.

[120] *See id.*

This modicum of creativity requirement may not be a problem if a brute-force content generation algorithm were designed to be creative. If an algorithm used "smart" brute-forcing[121] to intelligently generate a poem from a dictionary list of words using various evaluative sub-algorithms to determine the quality of a sentence or phrase, a modicum of creativity may be present.

Notwithstanding its creativity or lack thereof, is a brute-force content creation algorithm even an "author?"[122]

It doesn't take much to be an "author" of a copyrighted work. In *Burrow-Giles Lithographic Co. v. Sarony,*[123] the Supreme Court said that "author" meant someone "to whom anything owes its origin, originator, maker."[124] Following this loose definition, courts do not even require that an author physically fix the required creative expression herself: even a paralyzed author can be an "author" in American copyright law.[125]

It is thus generally assumed that, because a human is (usually) necessarily involved in the creation of computer code, computer-generated works including brute-force content creation algorithms are copyrightable, with some human being (such as the programmer or operator of the

---

[121] *See supra* Part II.D.

[122] 17 U.S.C. § 201 ("Copyright in a work protected under this title vests initially in the *author or authors* of the work.") (emphasis added).

[123] 111 U.S. 53, 4 S. Ct. 279, 28 L. Ed. 349 (1884).

[124] Burrow-Giles Lithographic Co. v. Sarony, 111 U.S. 53, 58, 4 S. Ct. 279, 281, 28 L. Ed. 349 (1884).

[125] This was discussed in *Fisher v. Klein,* 1990 WL 10072477 (S.D.N.Y. June 26, 1990):

> [U]nder the copyright law that authorship, even with respect to sculptors, need not be in the form of the manipulation of the material. [W]e had some discussion of the concept of a sculptor who might sit in a chair, never moving and never touching the materials, perhaps in part because he might be paralyzed or simply because the materials might be large and heavy. There are sculptors nowadays who work in huge materials, I-beams, storage tanks, things like that, that are welded together where the sculptor's contribution is rendered entirely by the giving of instructions to workmen to put a member in a certain position and bolt it to another member and so forth. I think it is clear without question that such participation in authorship. Such carrying out of ideas of authorship is recognized as authorship under the copyright law even if the author never places his hand on the material.

program) as an author.[126] Accordingly, the creator of an image who creates that image entirely using a computer image processing program (like Photoshop) is the author of that image for the purposes of copyright law. Even though some scholarship argues that the Patent and Copyright Clause may permit a non-human author (such as an artificial intelligence) to be the "author" of a copyrightable work,[127] it is unlikely that such a doctrine would be necessary when an algorithm is programmed and run under the supervision of a human being. This is especially the case with "smart" brute-forcing,[128] where a programmer's creativity and decisionmaking is more obviously present in the algorithm's programming. Assuming that a brute-force content creation algorithm was entirely independent and somehow did not involve the work of a programmer, this might be a different story[129] – however, it seems unlikely such an problem would ever arise, as even programming self-modifying code is quite a chore.[130]

Thus, brute-force content creation is almost by definition perfectly amenable to the doctrine of independent creation, and what it creates may be copyrightable if a modicum of creativity on the part of the algorithm is found.

But the law does not work in a vacuum. Just because something can generate copyrightable material does not mean that it would be found noninfringing.

---

[126] This is a grandiose oversimplification of the fascinating topic of computer-generated works. For an excellent, albeit slightly old, analysis of the issue of computer-generated works, including the issue of authorship vesting in an artificial intelligence, see Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?,* 106 HARV. L. REV. 977, 1042–72 (1993).

[127] *See id.*

[128] *See supra* Part II.D.

[129] *See generally* Arthur R. Miller, *Copyright Protection for Computer Programs, Databases, and Computer-Generated Works: Is Anything New Since CONTU?,* 106 HARV. L. REV. 977, 1042–72 (1993).

[130] On the topic of self-modifying code, see generally Hongxu Cai, Zhong Shao & Alexander Vaynberg, *Certified Self-Modifying Code,* PROC. 2007 ACM SIGPLAN CONF. ON PROGRAMMING LANGUAGE DESIGN AND IMPLEMENTATION (PLDI'07) 66–77, *available at* http://flint.cs.yale.edu/flint/publications/smc.pdf.

IV
LEGAL RAMIFICATIONS AND REALITIES OF A BRUTE-FORCED WORLD

Even if an entity actually attempted to brute-force copyright, and even if it managed to copyright what it made, it would not be legally immunized. In fact, the attempted creation or use of brute-forced creative material may be the first shot in an all-out copyright war.

Potential brute-force content generation business structures range the gamut from the nefarious to the benign. On one hand, a true profiteer could operate a number of brute-force content creation servers to find and sell all permutations that resembled existing works. On the other hand, a public interest organization dissatisfied with current copyright law could brute-force creative works to attack the entire concept of the ownership of creative works[131] by giving the public a free license to anything generated by the algorithm. Either business model could elect to search for valuable permutations itself or elect to allow third parties to search for valuable permutations they wished to purchase. Suffice to say, there are many permutations to the brute-forcing business model.[132]

That being said, with the act of brute-forcing content itself as the common denominator of any such business model, many things can be said about the legal ramifications of content brute-forcing: namely, that any attempt at brute-forcing copyright would almost certainly be found to infringe existing copyrights.

A. AN ALGORITHM DESIGNED TO INFRINGE(?)

[131] This organization could, for example, intentionally target works that would have gone into the public domain had the Copyright Term Extension Act (CTEA) and the Uruguay Round Agreements Act not gone into effect. *See* Golan v. Holder, 132 S. Ct. 873, 181 L. Ed. 2d 835 (2012); Eldred v. Ashcroft, 537 U.S. 186, 123 S. Ct. 769, 154 L. Ed. 2d 683 (2003); *see also* Lawrence Lessig, *How I Lost the Big One,* LEGAL AFF. (Mar./Apr. 2004), *available at* http://www.legalaffairs.org/issues/March-April-2004/story_lessig_marapr04.msp.

[132] I apologize for the terrible pun.

One infringes copyright by violating a copyright holder's exclusive rights.[133] Among many other rights, including those tailored to the nuances of specific creative works,[134] a copyright holder has the exclusive right to "reproduce [their] copyrighted work in copies."[135] These "copies" include "substantially similar" reproductions made by any means "now known or later developed, and from which the work can be perceived, reproduced, or otherwise communicated, either directly or with the aid of a machine or device."[136] Strictly speaking, this infringement inquiry does not rely upon whether or not the allegedly infringing materials can be copyrighted, as such a question comes *after* the question of whether or not such allegedly infringing materials infringe existing copyrights. If a court finds that allegedly infringing materials are independently created such that they can be copyrighted,[137] then it has already decided that such materials are not copies and thus cannot infringe.[138] Conversely, no independent creation means no copyright and potentially means infringement.[139]

If the content generated by a brute-force content creation algorithm was not amenable to the doctrine of independent creation, then the operator that algorithm would be in trouble: virtually every permutation generated by the algorithm could potentially infringe others' copyrights, as it would make both actual and substantially similar copies of copyrighted works in violation of numerous copyright holders' exclusive right to reproduce their work in copies.[140]

---

[133] *See* MERGES, MENELL & LEMLEY, *supra* note 88, at 518–19.

[134] For example, the owner of a dramatic work has the exclusive right to perform that work publicly. 17 U.S.C. § 106(5).

[135] 17 U.S.C. § 106(1); *see also* Arnstein v. Porter, 154 F.2d 464 (2d Cir. 1946).

[136] 17 U.S.C. § 101 (defining "copies").

[137] *See supra* Part III.A, III.B.

[138] Independent creation acts as both a defense to copyright infringement as well as a requirement for a copyright under the Copyright Act. 3 WILLIAM F. PATRY, PATRY ON COPYRIGHT § 9:32 (2012); 2 WILLIAM F. PATRY, PATRY ON COPYRIGHT § 3:30 (2012) (discussing both the requirement for copyright and the defense as substantially the same); *see also* Calhoun v. Lillenas Publ'g, 298 F.3d 1228, 1232–33 (11th Cir. 2002).

[139] As, again, independent creation is a statutory requirement under the Copyright Act. MERGES, MENELL & LEMLEY, *supra* note 88, at 421.

[140] *See* 17 U.S.C. § 106(1), 101.

That is, rather than independently creating many copyrighted works, the operator would effectively be potentially infringing *every single copyright in existence!*

But even if such permutations were found to be independently created, a court could still find that such permutations were infringing despite the limitations of modern copyright law. A court presented with a brute-force content creation algorithm could simply ignore the specific machinations of the algorithm, reducing its creation and operation to mere window-dressing around an attempt to infringe copyright in an intentionally obfuscated manner. After all, what judge would hold "independently created" copyrighted works generated from an algorithm valid when such a holding would facilitate a massive loophole around current copyright law and allow the mass creation of copies of copyrighted works?[141]

This less technical concept of infringement may seem like an extreme way to bend copyright law to punish seemingly "bad" behavior, but courts are no strangers to extending copyright law where they feel, for policy or other reasons, such an extension is warranted. The best example of courts extending copyright law in this way is the development of the law of contributory infringement. Unlike patent law,[142] the Copyright Act "does not expressly render anyone liable for infringement committed by another."[143] But copyright law nonetheless imposes liability for copyright infringement against "certain parties who have not themselves engaged in the infringing activity."[144] A court might similarly find extra-statutory infringement where a party intentionally sets up a brute-force content creation algorithm to create identical or substantially similar copies of copyrighted works, even if the operation of that algorithm may not infringe under any current legal doctrine.

---

[141] It could also essentially decimate the world of copyright. *See infra* Part IV.D.

[142] *See* 37 U.S.C. § 271(b), (c).

[143] Sony Corp. of Am. v. Universal City Studios, Inc., 464 U.S. 417, 434–35, 104 S. Ct. 774, 785, 78 L. Ed. 2d 574 (1984).

[144] *Id.*

But this is not the end of the analysis. A brute-force content creation algorithm can both infringe and not infringe copyright: after all, a brute-force content creation algorithm can generate already copyrighted material as well as wholly novel (and thus not copyrighted) material in the same second. Thus, brute-force content creation algorithms are amenable via analogy[145] to case law that involves products and devices that, like brute-force content generation algorithms, only sometimes infringe copyright.

The case law about devices that sometimes infringe emerges from a case involving a now-dead technology: videotape recorders. In *Sony Corp. of Am. v. Universal City Studios, Inc.,*[146] the Supreme Court held that the sale of equipment that could potentially duplicate copyrighted works (in this case, videotape recorders that could record live television) was not itself contributory copyright infringement:

> [T]he sale of copying equipment, like the sale of other articles of commerce, does not constitute contributory infringement if the product is widely used for legitimate, unobjectionable purposes. Indeed, it need merely be capable of substantial noninfringing uses.[147]

In other words, the Court held that the seller of a device that can be used to infringe is not liable for their purchasers' infringements as long as their device can be used for some other non-infringing purpose.[148] Subsequent courts have held that these "other purposes" – known as "substantial noninfringing uses" – need not even be actual, but merely capable, now or in the future.[149] This substantial noninfringing use doctrine does not, however, provide absolute immunity where actual infringement under the control of the device's creator takes place: courts often emphasize that computer system operators still have a duty to purge infringing material on

---

[145] "By analogy" because *Sony* and *Grokster* both are contributory infringement cases. As I discuss later in this subpart, this may be largely irrelevant.

[146] 464 U.S. 417, 104 S. Ct. 774, 78 L. Ed. 2d 574 (1984).

[147] Sony Corp. of Am. v. Universal City Studios, Inc., 464 U.S. 417, 442, 104 S. Ct. 774, 789, 78 L. Ed. 2d 574 (1984).

[148] *Id.*

[149] A&M Records, Inc. v. Napster, Inc., 239 F.3d 1004, 1020–21 (9th Cir. 2001).

their systems when they learn about it regardless of the substantial noninfringing uses their system may have.[150]

An important asterisk must be placed on the *Sony* decision: when courts smell bad intent on the part of a device's creator, they find liability even where the device in question has substantial noninfringing uses. In *Metro-Goldwyn-Mayer Studios Inc. v. Grokster Ltd.,* the Court acknowledged that the file-sharing services Grokster and Streamcast had potential noninfringing uses,[151] but nonetheless found that, because Grokster and Streamcast advertised and encouraged infringement on their services,[152] a court could find that those file-sharing services induced copyright infringement.[153] Specifically, the Court held that a party could be liable for copyright infringement by "distribut[ing] a device with the object of promoting its use to infringe copyright, as shown by clear expression or other affirmative steps taken to foster infringement."[154]

But there is an asterisk to the *Grokster* asterisk: this intent inquiry does not impose an affirmative duty upon a defendant to prevent copyright infringement. In footnote 12 of the *Grokster* opinion, the Court made a critical exception to its ruling:

> [I]n the absence of other evidence of intent, a court would be unable to find contributory infringement liability merely based on a failure to take affirmative steps to prevent infringement, if the device otherwise was capable of substantial noninfringing uses. Such a holding would tread too close to the *Sony* safe harbor.[155]

---

[150] *See, e.g., id.* at 1021.

[151] Though the Court nonetheless was of the view that the service(s) were primarily used for infringing copyright. Metro-Goldwyn-Mayer Studios Inc. v. Grokster, Ltd., 545 U.S. 913, 922–23, 125 S. Ct. 2764, 2772, 162 L. Ed. 2d 781 (2005) (The "vast majority of users' downloads [were] acts of infringement.").

[152] *Id.* at 925. Grokster and Streamcast advertised to former Napster users, encouraging their services as an alternative to Napster for (ostensibly illegal) file-sharing. *Id.*

[153] *Id.* at 941.

[154] *Id.* at 919.

[155] *Id.* at 939 fn. 12; *see also* Mark F. Schultz, *Will Bittorrent Go the Way of Grokster? File Sharing After* MGM v. Grokster, ABA SCITECH LAW., Winter 2006, at 4, 5.

Does a brute-force content creation algorithm have a substantial noninfringing use, now or in the future? The answer is unequivocally yes, at least in the abstract. But *Grokster* intent is a problem.

As already stated above, an algorithm designed to generate every possible permutation of text, images, or music is not specifically designed to infringe copyright – it generates both copyrighted and un-copyrighted material alike.[156] In the same second an image brute-forcing algorithm reproduces a copyrighted photograph, it may generate an aesthetically pleasing pattern that has never been created or even seen before. Thus, at least in the broadest sense, a brute-force content creation algorithm certainly has substantial noninfringing uses under *Sony.*[157]

But more realistically, a brute-force algorithm is valuable at least in part because it has the ability to independently create already copyrighted works. In other words, brute-force content creation algorithms exploit a loophole in copyright law. And that's where *Grokster* intent becomes a problem.

Assume a group of entrepreneurs with the time, money, and know-how to create an efficient and operative brute-force content creation business. As discussed above, such a business would require an astounding amount of time, money, and resources[158] – after all, a single one of Gosney's server clusters costs approximately $10,000.[159] It makes sense that this business would seek return on its expensive server investments, and one of the easiest ways to do this would be to sell independently created duplicates of existing copyrighted works.[160] This

---

[156] *See generally supra* Part II.
[157] *See* A&M Records, Inc. v. Napster, Inc., 239 F.3d 1004, 1020–21 (9th Cir. 2001).
[158] *See supra* Part II.C.
[159] *See id.*
[160] I am far from the first person to hypothesize this. Matt Kane, the man behind the random image generator "PixelMonkeys," wrote that his greatest fear about random image technology was "that some evil corporation someday will write an algorithm to increase the potential to create recognizable images and create a giant library of images." Matt Kane, *Frequently Asked Questions,* PIXELMONKEYS.COM, http://www.pixelmonkeys.org/#faq (last visited Mar. 13, 2013).

business model would require a remarkably low amount of effort on the part of the brute-forcing entity as, rather than spending the time and money sifting through permutation after permutation to ascertain market value by itself, the brute-forcing entity could create a search engine where potential purchasers could search through a collection of algorithm-generated works to find what they wanted to buy.[161] Needless to say, if the brute-forcing entity sold licenses to its permutations for less than copies of the original works cost,[162] the it could easily profit.

Accordingly, even though an otherwise infringing brute-force content creation algorithm may have substantial noninfringing uses, considering its use would almost certainly involve the nefarious intent to undermine copyright, the *Grokster* exception would almost certainly apply and the algorithm's substantial noninfringing uses would not provide a defense to infringement. Such an algorithm may not be itself commercially distributed as was the case in *Grokster,*[163] but this is almost certainly immaterial: *Sony* itself involved sale and distribution,[164] so it is unlikely that a defendant could invoke *Sony* as a defense without implicitly conceding that both *Sony* and *Grokster* apply when no sale or distribution occurs.

But assuming that the owners of the brute-force content generation algorithm were not nefarious profiteers, *Grokster*'s footnote 12 could provide a valuable safe harbor. If those utilizing a brute-force content creation algorithm did so not because of a desire to undermine copyright but instead because of a legitimate desire to produce new and unique works, then the mere fact that copyrighted work incidentally appeared on their storage devices would be

---

[161] The potential infringement ramifications of these searches are discussed *infra* in Part IV.B.

[162] An entity could arguably sell contractual "licenses" even if the permutations were not amenable to copyright. Contracts may legally bind parties to copyright-like terms even when the contractual *res* is not amenable to copyright protection. ProCD, Inc. v. Zeidenberg, 86 F.3d 1447, 1452–53 (7th Cir. 1996). That being said, such contracts would not prevent other businesses from copying the permutations –they would only bind the parties involved.

[163] *See* Metro-Goldwyn-Mayer Studios Inc. v. Grokster, Ltd., 545 U.S. 913, 919, 125 S. Ct. 2764, 2770, 162 L. Ed. 2d 781 (2005).

[164] *See* Sony Corp. of Am. v. Universal City Studios, Inc., 464 U.S. 417, 442, 104 S. Ct. 774, 789, 78 L. Ed. 2d 574 (1984).

inconsequential.[165] Such do-gooders would not have an affirmative duty to sift through every permutation and delete infringing content to act within the ambit of copyright law.[166] Unfortunately, these do-gooders would likely be forced to sift through every permutation anyway to find something useful in a massive database of algorithm-generated nonsense.

Thus, the operator of a brute-force content creation algorithm would in most circumstances be infringing many copyrights, if not potentially every copyright in existence. Even though the *Sony* line of cases might appear to immunize brute-forcing behavior by analogy, those cases would not provide a defense to those most likely to operate a brute-force content creation algorithm: profiteers looking to generate plausibly legal copies of existing copyrighted works. Simply put, a brute-force content creation algorithm is one massive infringement case waiting to happen, even if a court has to proverbially bend over backwards to make it such.

B. THE INDUCEMENT PROBLEM

A brute-force content creation algorithm could also infringe copyright depending on the way third parties use brute-forced permutations. This could occur in two ways: first, by merely allowing parties to search through the permutations themselves, and second, by selling or giving parties permutations to enable those third parties to infringe.

As illustrated in Part I,[167] a brute-forcing algorithm can easily generate a huge number of permutations. This is a problem: manually sifting through permutations to find something valuable (such as an exact duplicate of another's copyrighted work) would be prohibitively expensive and time-consuming, especially considering how reasonable minds could differ as to the artistic merit of any given permutation.

---

[165] *See* Metro-Goldwyn-Mayer Studios Inc. v. Grokster, Ltd., 545 U.S. 913, 939 fn. 12, 125 S. Ct. 2764, 162 L. Ed. 2d 781 (2005).

[166] *See id.* at 939 fn. 12; *see also* Mark F. Schultz, *Will Bittorrent Go the Way of Grokster? File Sharing After* MGM v. Grokster, ABA SCITECH LAW., Winter 2006, at 4, 5.

[167] *See supra* Part II.

An easier way to search through permutations would be to use a computer system to find a duplicate of an already existing copyrighted work within the permutations,[168] but this raises an infringement issue. Assuming that merely storing and generating brute-forced permutations did not constitute infringement in and of itself,[169] could searching through these permutations to find a copyrighted work constitute a form of infringement by either the brute-forcing entity or a third party? If a third party used an existing copyrighted work (like an image) to find a duplicate of it in a brute-force content collection, would that act constitute infringement?

The mere indexing and searching of content is not, in and of itself, an infringement of copyright. In *Perfect 10, Inc. v. Amazon.com, Inc.,*[170] Perfect 10 sued Google and Amazon.com for copyright infringement because both defendants allowed users to search for Perfect 10 material stored on third party websites.[171] Because neither defendant stored the full versions of Perfect 10's photographs,[172] the court found that the defendants did not infringe Perfect 10's display right as to the full versions of the works.[173] In other words, only where storage of infringing works occurs will liability attach.[174] *Perfect 10* indicates that, absent some other form of infringement involving the subject material, making indexing and searching of non-infringing material feasible is not independently a violation of copyright.

---

[168] Google already does something similar with their Google Images search engine: a user can upload any digital image and be given results based on the image, including places in which it (or images very similar to it) are located online. *See Search by Image*, GOOGLE.COM, http://www.google.com/insidesearch/features/images/searchbyimage.html (last visited Mar. 23, 2013). In a brute-force content creation database, this could be achieved using MD5 checksum algorithms, which can be used to compare everything from images to bioinformatics sequence identifiers. *See, e.g.,* Mike Smith et al., *MagicMatch – Cross-Referencing Sequence Identifiers Across Databases,* 21:16 BIOINFORMATICS 3429–30 (June 16, 2005), *available at* http://bioinformatics.oxfordjournals.org/content/21/16/3429.full.

[169] An unlikely circumstance, given how likely such actions alone would be infringing. *See supra* Part IV.A.

[170] 508 F.3d 1146 (9th Cir. 2007).

[171] Perfect 10, Inc. v. Amazon.com, Inc., 508 F.3d 1146, 1154 (9th Cir. 2007).

[172] Google did, however, store thumbnails, which were found to be protected by fair use. *Id.* at 1160, 1163–68.

[173] *Id.* at 1159–63.

[174] *See id.*

Thus, where infringement arguments based upon the storage of brute-force generated content fail, the aforementioned search argument would fail as well. While there might be an argument that the use of an existing copyrighted work as a referent in the process of searching for a duplicate of it is a form of infringement, this does not seem to be a particularly fruitful argument. Given the strength of the underlying infringement argument discussed above,[175] this is not truly a major loss for a would-be plaintiff.

Perhaps more important is the question of third party infringement by using brute-forced permutations as a substitute for copyrighted material. While the concept of secondary liability in copyright is "muddied,"[176] it is generally accepted that one who, "with knowledge of the infringing activity, induces, causes or materially contributes to the infringing conduct of another, may be held liable as a 'contributory' infringer."[177] If brute-forced permutations are not amenable to copyright,[178] the provision of those permutations to third parties to enable those parties to indirectly acquire existing copyrighted works without acquiring a license to those copyrighted works would almost certainly be a form of contributory infringement. After all, part of the value of brute-forced permutations would be their similarity to existing copyrighted work, and the purchaser of such a low-cost permutation would almost certainly not possess a license to the original work. Selling those permutations would be little better than selling pirated copies of a copyrighted movie on the Internet.

---

[175] *See supra* Part IV.A.

[176] 6 WILLIAM F. PATRY, PATRY ON COPYRIGHT § 21:41 (2012) (discussing how, in *Sony*, the Supreme Court used the doctrines of contributory infringement and vicarious liability interchangeably).

[177] Gershwin Pub. Corp. v. Columbia Artists Mgmt., Inc., 443 F.2d 1159, 1162 (2d Cir. 1971); *see also* Metro-Goldwyn-Mayer Studios Inc. v. Grokster, Ltd., 545 U.S. 913, 914, 125 S. Ct. 2764, 2767, 162 L. Ed. 2d 781 (2005).

[178] As is almost certainly the case. *See supra* Part III. Note that if they were copyrightable, then license to the generated permutations could arguably be given, like Learned Hand's *Ode on a Grecian Urn* example. *See supra* Part III.A.

A brute-force content creation entity could potentially limit its contributory infringement liability under this scenario by carefully contracting with the third party buyers of its permutations. In *ProCD, Inc. v. Zeidenberg,* the Court of Appeals for the Seventh Circuit allowed a software manufacturer to enforce so-called "shrink-wrap licenses" – that is, licenses included with software that placed additional limitations on the use of that software – that extended protection of its works beyond the scope afforded to them under the Copyright Act.[179] The court in *Zeidenberg* allowed these shrink-wrap licenses because "[c]ontracts . . . generally affect only their parties; strangers may do as they please, so contracts do not create 'exclusive rights.'"[180] *ProCD* thus seems to embrace the idea that a brute-force content generating entity could create and enforce shrink-wrap licenses (or their equivalent) to prevent third-parties from using a permutation without a license from the original copyright holder, thereby potentially avoiding contributory infringement liability. Given the controversy of *ProCD,*[181] it may not be the case that such a contract would be upheld in every court, but shrink wrap licenses are nonetheless an option for an already risk-taking brute-forcing entity.

C. THE TRADEMARK DIMENSION

Brute-force content creation would almost certainly entail the mass replication of both registered and unregistered trademarks. This could make the content produced by a brute-force content creation algorithm a trademark infringement landmine. But, perhaps thankfully, because permutations are unlikely to create consumer confusion, it is unlikely that a brute-force content creation algorithm would ever be found to infringe trademark.

---

[179] *See* ProCD, Inc. v. Zeidenberg, 86 F.3d 1447, 1452–53 (7th Cir. 1996). These contracts extended beyond the scope of copyright because the underlying material protected in the *ProCD* contract was unprotectable by copyright post-*Feist. See id.*

[180] *Id.* at 1454.

[181] *See generally* David Rice, *Copyright and Contract: Preemption after* Bowers v. Baystate, 9 ROGER WILLIAMS U. L. REV. 595, 610–613 (2004); Brian Covotta & Pamela Sergreef, ProCD, Inc. v. Zeidenberg, 13 BERKELEY TECH. L. J. 35 (1998); 6 WILLIAM F. PATRY, PATRY ON COPYRIGHT § 18:26 (2012).

"Trademark" includes any word, name, symbol, device, or combination used by a person to identify and distinguish his or her goods, including a unique product, from those manufactured or sold by others.[182] A trademark must be inherently distinctive – that is, unique and immediately identifiable as identifying a unique product source – or must have "secondary meaning" – which means the mark must have an established connection with a unique product source.[183] Once these requirements are established, when a mark is federally registered, the mere act of using that mark in connection with the sale or advertising of goods or services such that the use would "likely . . . cause confusion" is infringement of that mark.[184] When a mark is unregistered, the use must cause confusion, mistake, or deception as to the "affiliation, connection, or association" of the entity with the mark's owner to constitute infringement.[185] In either case, there is no independent creation defense in trademark law.[186]

Arguably, the sale or even provision of brute-forced content could be considered a form of trademark infringement. For example, an image brute-forcing algorithm might generate an image permutation that depicted the Nike logo or an audio permutation that used the trademarked slogan "King of Beers."[187] As the argument might go, because trademark law has no independent creation doctrine,[188] the sale or use of such a permutation could constitute an infringing use of that trademark.

Unfortunately for a would-be plaintiff, it is not clear whether consumers would be actually confused by a brute-force content generation algorithm's use of a mark such that the

---

[182] Lanham Act § 45, 15 U.S.C. § 1127.

[183] *See* Zatarains, Inc. v. Oak Grove Smokehouse, Inc., 698 F.2d 786 (5th Cir. 1983); MERGES, MENELL & LEMLEY, *supra* note 88, at 751.

[184] Lanham Act § 32(1), 15 U.S.C. § 1114(1).

[185] Lanham Act § 43(a)(1)(A), 15 U.S.C. § 1125(a)(1)(A).

[186] Douglas Y'Barbo, *The Heart of the Matter: The Property Right Conferred by Copyright,* 49 MERCER L. REV. 643, 683 (1998).

[187] "King of Beers" is a trademarked slogan used by Budweiser.

[188] *Id.*

existence of that mark in a permutation would be infringing. With the right disclaimers, a brute-forcing entity could avoid creating consumer confusion in the way that would expose itself to liability for infringement.[189] But such a disclaimer may not be necessary. In *Medic Alert Foundation U.S., Inc. v. Corel Corp.,*[190] the court held that the presence of a logo in a software clipart library collection was not trademark infringement because users would not have been confused into believing that the owner of each respective mark was endorsing the defendant's collection.[191] A brute-forced content collection is arguably analogous to a collection of clipart: neither use trademarks in a way that implicate association, sponsorship, or any affiliation such that consumers would be confused. This is especially the case where consumers know about the nature of the permutations: no rational consumer would believe that an algorithmically generated permutation was intentionally designed to affiliate with a unique product source. Thus, it is unlikely that a court would find consumer confusion if a trademark was generated by a brute-force content creation algorithm.

A more viable approach might be the argument of dilution by blurring, but *Medic Alert*-like issues still apply. Where a famous mark is used by a party in a way that could potentially dilute the potency of that mark (by reducing its ability to identify a single source and maintain selling power or the like), the owner of that mark may sue.[192] The test for dilution by blurring involves a number of factors, including whether the user of the mark intended to create an association with the famous mark and whether there was any actual association between the allegedly infringing mark and the famous mark.[193] Assuming some famous mark (such as the

---

[189] *See* Lanham Act § 43(a)(1)(A), 15 U.S.C. § 1125(a)(1)(A); Lanham Act § 32(1), 15 U.S.C. § 1114(1); Louis Vuitton Malletier S.A. v. Haute Diggity Dog, LLC, 507 F.3d 252 (4th Cir. 2007).

[190] 43 F. Supp. 2d 933, 51 U.S.P.Q.2d 1024 (N.D. Ill. 1999).

[191] Medic Alert Found. United States, Inc. v. Corel Corp., 43 F. Supp. 2d 933, 938, 51 U.S.P.Q.2d 1024, 1028 (N.D. Ill. 1999).

[192] Lanham Act § 43(c), 15 U.S.C. § 1125(c).

[193] Lanham Act § 43(c)(2)(b)(v)-(vi), 15 U.S.C. § 1125(c) (2)(b)(v)-(vi).

Nike swoosh) made its way into a permutation, trademark infringement would plausibly exist. But the *Medic Alert* problem still exists for a would-be plaintiff. Given that the nature of the use of a famous mark is evaluated in a case of dilution by blurring,[194] a court could plausibly find that, like in *Medic Alert,* the incidental use of a mark in a database that was unlikely to cause consumer confusion would not qualify as dilution by blurring.

One factor that almost certainly does not influence either trademark infringement calculus is whether or not the brute-forcing entity purported to provide its brute-forced content for free. Non-commercial use is a defense to trademark infringement, though it has never been entirely clear what non-commercial use entails.[195] Thus, at least theoretically, if a brute-forcing entity provided its work for free, it could avoid being liable for trademark infringement. But this result is only theoretical: it ignores the fact that, even if it provided its content for free, a non-commercial entity would usually seek to provide its content for free to directly undermine or manipulate the market for copyrighted materials (unless, of course, it was simply hunting for unique patterns or novel permutations or the like).[196] This sort of behavior is quite unlike the archetypal non-commercial trademark user who, for example, uses a trademark to complain about a company.[197] Thus, even though infringement is unlikely to be proven, non-commercial use of brute-forced permutations would not provide a defense would brute-force content creation infringe trademark.

---

[194] *Id.*

[195] *See generally* 4 J. THOMAS MCCARTHY, MCCARTHY ON TRADEMARKS AND UNFAIR COMPETITION § 24:128 (4th ed. 2013); Lee Ann W. Lockridge, *When Is A Use in Commerce A Noncommercial Use?,* 37 FLA. ST. U. L. REV. 337, 390 (2010) (arguing that "[a] use in commerce is a noncommercial use when the use of a mark either intertwines commercial and noncommercial speech elements or is not an integral part of a commercial transaction, i.e., when the use is not purely commercial speech.").

[196] *See supra* Part IV.I (discussing the *Grokster* footnote 12 exception).

[197] *See, e.g.,* Smith v. Wal-Mart Stores, Inc., 537 F. Supp. 2d 1302, 1309 (N.D. Ga. 2008).

D. COPYRIGHT WAR

The idea that any entity could brute-force copyright is positively frightening. If a single entity could construct an algorithm that made every possible 4-minute song ever, then that entity would have the power to sell exact copies of other artists' 4-minute songs for cheaper than the artists sold them. The incentive to create could become a *dis*incentive, as musicians would suddenly find even the most original songs they could ever compose anticipated by a computer algorithm that could undercut their profits.

In loftier constitutional terms, a brute-force content creation algorithm would prevent copyright from "promot[ing] the Progress of Science."[198]

Or, more bluntly, it would kill copyright.

Suffice to say, it is unlikely that a court would ever allow this to happen in the first place. Before such a scheme could ever begin, it is almost certain that a court would stop a brute-force content creator through one or more methods – be it a strict "modicum of creativity" standard,[199] an attenuated standard of infringement,[200] or the like. This is especially the situation where, in most cases, *Grokster* intent would weigh against the operator of a brute-force content creation algorithm.[201]

But assuming through some apocalyptic judicial catastrophe that the above arguments failed, would a party operating a brute-force content generation algorithm win? Once that party gets past the hurdles of skirting the copyright and trademark infringement issues discussed above, are they legally home free?

---

[198] U.S. CONST. art. I, § 8, cl.8. "Science" in this context refers to creative works.
[199] *See supra* Part III.B.
[200] *See supra* Part IV.A.
[201] *See id.*

The answer is almost certainly no, because such an algorithm would kill copyright. The fact that a brute-force content creation algorithm would in effect give a copyright in every possible work ever to a single party would give every single judge, legislator, and citizen a reason to specifically prohibit that from happening. When someone finds a loophole in the law that grants them a legal right to every creative work that could ever be created, the societal response would never be "you win" – it would be "we need to amend the laws."

In other words, operating a brute-force content creation algorithm would be the first shot in all-out copyright war between a single entity and the entire legal and political community of the United States, if not the world. It is not hard to imagine who would win.

V

CONCLUSION: WHY "NEVER" IS A GOOD THING

As has been repeatedly emphasized in this paper, it is almost certain that it is technically[202] and legally[203] impossible for anyone to brute-force copyright in the near future. This is, unquestionably, a very good thing.

Imagine a brute-forced collection of four-minute sound recordings. What would be within those recordings? Not just songs, but human experiences. Four minute conversations, laughs, cries, speeches – quite literally *everything*. The whole of the human experience that could be heard would, insofar as it fits into a four minute digital recording, be located on computer disks squirreled away within a huge series of processing servers. This would include everything you have ever heard, as well as everything you will ever hear. This is no less the case for an attempted brute-force attack on images the size of a Google logo – everything that could

[202] *See supra* Part II.C.
[203] *See supra* Parts III, IV.

ever be seen that could be represented in a small bundle of pixels on a monitor would be generated and stored.

As disappointing as it may be that current technology cannot yet brute-force creative content,[204] this is actually a good thing: it proves how amazingly diverse and unique the world can be. If it were easy to simply brute-force through every song ever made, then the actual number of songs that could ever be made would be rather small.[205] There is no romance or magic to a world of creative content that can be divided, processed, and in the end conquered by an emotionless machine designed to feign creative activity.

Of course, creative activity does not rely upon copyright, and if copyright were to disappear, many authors would still create amazing works. But a brute-force content creation attack is not merely a war upon copyright: it is a war on creation. It is an attempt to preclude anyone from creating anything truly new ever again, even if the algorithm itself never uses the work it generates.

This is why, somewhat counter-intuitively, we should hope that technology is never able to process the insane number of permutations discussed in Part II.[206] When all songs, paintings, and poems have been generated, then the desire for artists to pick up their respective guitars, paint brushes, and pens will be inhibited, if not entirely destroyed, and part of the enjoyment that arises from creative material – that is, the knowledge that an individual or group individuals poured their lives and selves into a project for others' enjoyment – would be decimated.

Thus, my discussion ends not with a legal conclusion, but a normative one: the world does not need brute-force content creation algorithms. In an attempt to make money, someone

---

[204] *See supra* Part II.C.

[205] Of course, "rather small" in the sense that one of Gosney's servers could have eventually brute-forced it at 350 billion guesses per second. *See supra* Part II.C.

[206] *See supra* Part II.

running such an algorithm would not merely kill copyright – they would kill the entire human drive to create around which copyright laws have formed.