Harvard Journal of Law & Technology
Volume 32, Number 2 Spring 2019

APPLICATION OF ZERO-KNOWLEDGE PROOF IN RESOLVING DISPUTES OF PRIVILEGED DOCUMENTS IN E-DISCOVERY

*Yuqing Cui**

TABLE OF CONTENTS

I. INTRODUCTION

In recent years, as e-discovery of electronically stored information ("ESI") has become widely adopted, the number of disputes over privileged documents have also exploded. Resolving these disputes in large civil cases often involves lengthy court adjudications, *in camera* reviews, and sometimes even special masters appointments to oversee the

* Yuqing Cui is a 3L at Harvard Law School. She obtained her Ph.D. in Chemical Engineering from MIT in 2016. Special thanks to Professor Martha Minow for her helpful comments and insights.

process.[1] As one judge put it, "such a situation is detrimental to the litigants, the courts, and our system of justice."[2] In addition to the sheer amount of work involved, judges are also tasked with striking the delicate balance between imposing high financial costs on the privilege-claiming party by demanding detailed descriptions of the claimed documents in the privilege logs,[3] and risking allowing non-privileged documents to be unfairly withheld.[4] As a result, privilege disputes have become a vexing legal problem. They await better solutions.

At the core of the disputes surrounding privileged documents is a simple trust problem: the privilege-claiming party holds secret documents that it is unwilling to show to the requesting party, who suspects the veracity of the privilege-claim. In other words, the privilege-claiming party wants to prove that the documents are indeed privileged without disclosing the documents' contents. This is, in fact, a classical problem that can be solved by a cryptographic concept called zero-knowledge proof.

Zero-knowledge proof has a seemingly contradictory definition: to be successful, a protocol needs to convince the verifier of the veracity of a statement without revealing the content supporting that statement. For example, if two children, Alice and Bob, want to see if they have received the same number of Halloween candies without showing each other their respective candy collections, they can use the following zero-knowledge proof implementation. Bob can label each of four locked boxes with different numbers. Only one box will be labeled with the number of candies that Bob has. He will keep the key to that box and will throw away the keys to all the other boxes. Alice will then slip identical pieces of paper into each box. If Alice sees a box labeled with the number of candies she holds, she will place a special mark on the paper she places in that box. If Bob then opens up the only box he has a key to, and sees the special mark, Alice and Bob will know they have

1. *See, e.g.*, *In re* Vioxx Prods. Liab. Litig. (*In re Vioxx*), 501 F. Supp. 2d 789, 791–92 (E.D. La. 2007) (for a detailed discussion, see *infra* Part II); Blair Harrington, *The Power of the Privilege Log*, MINN. STATE BAR ASS'N: BENCH & BAR OF MINN. (June 3, 2018), http://mnbenchbar.com/2018/06/the-power-of-the-privilege-log [https://perma.cc/Z56F-PY82] (discussing the process of handling privilege log disputes and noting that "[i]f not adequately addressed, privilege logs can become a major roadblock during discovery"). For a detailed discussion, see *infra* Part II.

2. *In re Vioxx*, 501 F. Supp. 2d at 815.

3. *See* Bryan Corp. v. Chemwerth, Inc., 296 F.R.D. 31, 41 (D. Mass. 2013) (stating that privilege logs need not be "precise to the point of pedantry," but instead need only reasonably describe the materials being withheld); Phillips v. C.R. Bard, Inc., 290 F.R.D. 615, 637–38 (D. Nev. 2013) (discussing the need for flexibility).

4. *See, e.g.*, Chevron Corp. v. Weinberg Grp., 286 F.R.D. 95, 98 (D.D.C. 2012) ("[T]he intendment to the Rule is clear: the opposing party should be able, from the entry to the log itself, to assess whether the claim of privilege is valid.").

the same number of candies; otherwise they will know they have different amounts of candies.[5] See Figure 1 for a visual representation of this scenario. Zero-knowledge proofs also serve a role in business and industry, such as acting as escrow agents in financial transactions, or calculating whether a salesperson has remitted appropriate taxes from her sales to be paid by a counterparty, without revealing the precise amount for which she was able to sell an item.[6]

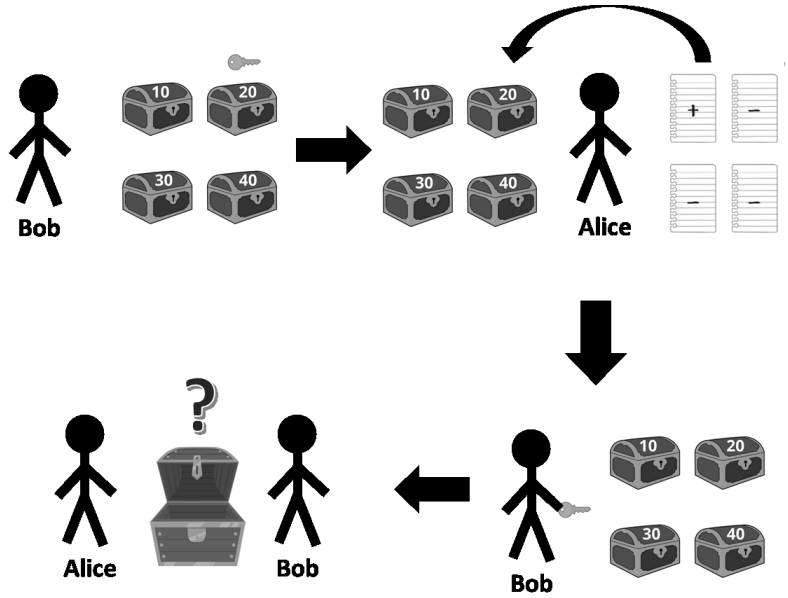

Figure 1: Zero-Knowledge Protocol for the Alice/Bob Candy Scenario[7]

Zero-knowledge proof is an active research area. Its applications in law have only recently begun to attract attention. Joshua Kroll contemplated applying zero-knowledge protocols to ensure that decision-makers or machine learning algorithms apply policies consistently across all decision subjects.[8] These policies could concern voting, approving loan and credit card applications, targeting citizens or neighborhoods for police scrutiny, setting bail or parole, selecting taxpayers for IRS audits, and granting or denying immigration visas.[9]

---

5 . Cossack Labs, *Explain Like I'm 5: Zero Knowledge Proof (Halloween Edition)*, HACKERNOON (Oct. 26, 2017), https://hackernoon.com/eli5-zero-knowledge-proof-78a276db9eff (last visited May 11, 2019).

6. *See* Joshua A. Kroll et al., *Accountable Algorithms*, 165 U. PA. L. REV. 633, 668 (2016).

7. Figure 1 adapted from Cossack Labs, *supra* note 5.

8. Kroll et al., *supra* note 6, at 682.

9. *Id.* at 636.

Besides Kroll's proposal, there are no other prominent application of zero-knowledge proof in the legal context. This Note focuses on the concept of zero-knowledge proof, describes the parallels between the problems it solves and the problems with disputes surrounding privileged documents, and illustrates that there are opportunities for the application of zero-knowledge proof in the broader legal context.

Part II of this Note discusses a specific legal issue that is prevalent in civil litigation — disputes of privileged documents. In the age of e-discovery these disputes have become numerous and burdensome for all parties involved.[10] One of the problems which arises in these disputes is the lack of trust between the parties about their claimed privileges.[11] In cryptography, such distrust problems can be solved with zero-knowledge proof. Part III explains this concept using a few examples. Part IV proposes two solutions to disputes surrounding privileged documents modeled on examples of zero-knowledge proof. The first solution involves applying a machine learning algorithm to identify privileged documents. The machine learning algorithm can either be trained with case-specific documents or with privileged documents of a specific type from a vast pool of cases. Under this solution, special care needs to be taken to ensure transparency and trust-building between opposing parties. The second solution involves masking keywords and concepts to mitigate the risk of disclosing potentially sensitive content in privilege challenges. Part V concludes the Note.

II. DISPUTES SURROUNDING PRIVILEGED DOCUMENTS PRESENT A CUMBERSOME TRUST PROBLEM FOR ALL PARTIES INVOLVED

During discovery for civil litigation in federal court, a party is under a legal duty to disclose certain information requested by the opposing party.[12] A party may withhold responsive information from a production request on the basis of privilege,[13] but that party typically

10. *See* Henry S. Noyes, *Good Cause Is Bad Medicine for the New E-Discovery Rules*, 21 HARV. J.L. & TECH. 49, 51 (2007) (noting that e-discovery is "more time-consuming, more burdensome, and more costly than conventional discovery").

11. *See, e.g.*, Eureka Fin. Corp. v. Hartford Accident & Indem. Co., 136 F.R.D. 179, 182–83 (E.D. Cal. 1991) (finding that "a general objection to an entire discovery document on the basis of privilege" is improper because it would require an opposing party to "simply trust the good faith and diligence of the party asserting the privilege").

12. *See* FED. R. CIV. P. 26.

13. FED. R. CIV. P. 26(b)(1) (distinguishing between "privileged" and "nonprivileged" matters in the scope of discovery, specifying that "[p]arties may obtain discovery regarding any *nonprivileged* matter that is relevant to any party's claim or defense" (emphasis added)).

should create a "privilege log" identifying what privileged information is being withheld.[14]

Although Rule 26 of the Federal Rules of Civil Procedure has long governed the discovery process in general, privilege logs were governed by local rules or by judge orders on a case-by-case basis prior to the enactment of Rule 26(b)(5) in 1993.[15] Today, when "information produced in discovery is subject to a claim of privilege or of protection as trial-preparation material," Rule 26(b)(5) requires a party to "notify any party that received the information of the claim and the basis for it."[16] Since the rule's enactment, it has become customary for the privilege-claiming party to comply with Rule 26(b)(5) by producing a privilege log for each document, containing enough information for the court or the opposing party to assess the claim of privilege.[17]

Rule 26(b)(5) deliberately left out the details on how to make a claim of privilege or work-product protection and what information is needed to justify such claims.[18] This is because claims of privilege often come in different forms, and thus appropriate justifications vary depending on case-specific circumstances.[19] But "the absence of explicit guidance as to the nature of the required [information] enlarges the vacuum in which strategic manipulation of the discovery process . . . may flourish."[20] Courts must strike a delicate balance between the request for information to establish the privilege claims and the burden such requests put on the privilege-claiming party; although blanket asser-

---

14. Michael Downey & Paige Tungate, *Practical Advice on Privilege Logs*, ABA L. PRAC. TODAY (Sept. 14, 2018), https://www.lawpracticetoday.org/article/practical-advice-privilege-logs [https://perma.cc/SK9B-XERR].

15. For a comprehensive review of the history of privilege logs prior to 1993, see John M. Facciola & Jonathan M. Redgrave, *Asserting and Challenging Privilege Claims in Modern Litigation: The Facciola-Redgrave Framework*, 4 FED. CTS. L. REV. 19, 23–27 (2010) (reviewing the inconsistencies surrounding what constituted an adequate log and the consequences of failing to live up to those standards prior to the enactment of FED. R. CIV. P. 26(b)(5)).

16. FED. R. CIV. P. 26(b)(5).

17. *See, e.g.*, Garcia v. E.J. Amusements of N.H., Inc., 89 F. Supp. 3d 211, 215 (D. Mass. 2015) ("The universally accepted means of claiming that documents are privileged is the production of a privilege log." (internal quotation marks omitted)); Acosta v. Target Corp., 281 F.R.D. 314, 320 (N.D. Ill. 2012) ("Traditionally, in this district that has been done by serving a privilege log [setting forth the required information]."); Caudle v. District of Columbia, 263 F.R.D. 29, 35 (D.D.C. 2009) ("A privilege log has become an almost universal method of asserting privilege under the Federal Rules."); SPX Corp. v. Bartec USA, LLC, 247 F.R.D. 516, 527 (E.D. Mich. 2008) ("[A] privilege log is customarily provided.").

18. *See* FED. R. CIV. P. 26 advisory committee's note to 1993 amendment ("The rule does not attempt to define for each case what information must be provided when a party asserts a claim of privilege or work-product protection.").

19. *See Garcia*, 89 F. Supp. 3d at 215.

20. Burlington N. & Santa Fe Ry. Co. v. U.S. Dist. Court, 408 F.3d 1142, 1148 (9th Cir. 2005). *See generally* Rebecca A. Cochran, *Evaluating Federal Rule of Civil Procedure 26(b)(5) as a Response to Silent and Functionally Silent Privilege Claims*, 13 REV. LITIG. 219 (1994) (describing forms of privilege claim abuse).

tions of privilege are inadequate to satisfy the claiming party's burden,[21] requiring too much description risks giving away privileged information and increases the cost and burden of preparing privilege logs.[22]

Even when a party meticulously prepares a document-by-document privilege log, few litigators are willing to trust "an opponent's understanding of the law and willingness to be forthcoming" in their determination of privileged documents.[23] When disputes arise in this context, courts use *in camera* review or special masters to review privileged materials, both of which are costly and time-consuming.[24]

The advent of e-discovery (i.e. electronic discovery of ESI) further exacerbated the already complicated issues with privilege logs. The primary challenge brought by e-discovery is the volume of privileged documents that need to be logged and described. One case, *In re Vioxx Products Liability Litigation*,[25] illustrates this challenge: a large volume of documents coupled with ambiguous guidance on the preparation of privilege logs.[26] In *Vioxx*, defendant Merck produced over two million documents amounting to eighteen million pages of documents in response to a discovery request.[27] Merck claimed privilege for one percent of the documents.[28] The district court ordered Merck to submit for *in camera* review all documents to which Merck claimed privilege.[29] In response to the order, Merck delivered eighty-one boxes "containing approximately 30,000 documents, amounting to nearly 500,000 pages."[30] The district judge "undertook the herculean task of personally reviewing 30,000 documents over a two-week period," but ended up with inconsistent results, concluding that one copy of a document was privileged and that exact duplicates of the same document

21. *See, e.g.*, *Burlington*, 408 F.3d at 1149 (holding that "boilerplate objections or blanket refusals inserted into a response to a Rule 34 request for production of documents are insufficient to assert a privilege"); Johnson v. Gross, 611 Fed. Appx. 544, 547 (11th Cir. 2015) (noting that the party who told opposing counsel that it was "not [his] job" to "parse it out" was "incorrect on the law").

22. *See, e.g.*, Facciola & Redgrave, *supra* note 15, at 32 n.55 (detailing that in an experiment conducted by the author-judge, many lawyer-participants felt that the privilege log description — a "[l]etter providing legal advice as to tax consequences of the proposed Smith deal" — provided too much information such that there was real risk of privilege waiver).

23. Downey & Tungate, *supra* note 14.

24. *See, e.g.*, NLRB v. Interbake Foods, LLC, 637 F.3d 492, 502 (4th Cir. 2011) (holding that if a privilege log discloses prima facie basis for privilege, the party seeking *in camera* review must identify facts reasonably suggesting that the materials are not in fact privileged); Lurensky v. Wellinghoff, 271 F.R.D. 345, 355–56 (D.D.C. 2010) (noting that *in camera* review should be exception rather than rule due to burden placed on the court).

25. *In re* Vioxx Prods. Liab. Litig. (*In re Vioxx*), 501 F. Supp. 2d 789 (E.D. La. 2007).

26. *Id.* at 790–93.

27. Vioxx Prods. Liab. Litig. Steering Comm. v. Merck & Co. (*Vioxx v. Merck*), No. 06-30378, 06-30379, 2006 WL 1726675, at *1 (5th Cir. May 26, 2006).

28. *Id.*

29. *In re Vioxx*, 501 F. Supp. 2d at 791.

30. *Id.*

were not.[31] While commending the district judge's efforts, the Fifth Circuit suggested that the district court instead sample only representative documents.[32] Merck duly provided the district court with another ten boxes of 2,000 documents.[33] The district court appointed two special masters to review these documents and 600 additional documents offered by the plaintiffs from the privilege log.[34] The second round of court review took three months and cost $400,000.[35] Eventually, the court ordered Merck to produce documents in accordance with guidelines produced in the special masters' report.[36] *Vioxx*'s saga highlights the urgent need for a better way to resolve privilege log disputes in the discovery of ESI.[37]

At its core, privilege log disputes can be distilled down to the following problem: the privilege-claiming party wants to prove to the opposing party that it indeed possesses privileged documents, without conveying any substantive information. This in fact is the exact challenge that zero-knowledge proof, an increasingly popular concept in cryptography, is designed to solve.

III. ZERO-KNOWLEDGE PROOF ENABLES VALIDATION OF A STATEMENT WITHOUT REVEALING ANY OTHER INFORMATION

Zero-knowledge proof is a cryptographic tool that makes it possible for a prover (here, the party attempting to prove privilege) to convince a verifier of the prover's knowledge of an assertion without revealing any information other than the validity of the prover's assertion.[38] Put simply, zero-knowledge proof is a method to validate a statement by revealing only the veracity, and nothing more. First introduced by Goldwasser et al. in 1985,[39] zero-knowledge proof has become an

31. *Vioxx v. Merck*, 2006 WL 1726675, at *2.
32. *Id.* at *10.
33. *In re Vioxx*, 501 F. Supp. 2d at 791.
34. *Id.* at 791–92.
35. *Id.* at 815 n.35.
36. *Id.* at 815–16.
37. Scholars have been trying to solve this problem using traditional legal methods. *See, e.g.*, Facciola & Redgrave, *supra* note 15, at 19–20 (proposing a framework where, instead of "traditional document-by-document privilege log," counsels cooperate under "early, careful, and rigorous judicial" supervision).
38. Li Feng & Bruce McMillin, *A Survey on Zero-Knowledge Proofs*, 94 ADVANCES IN COMPUTERS 25, 25–69 (2014). Zero-knowledge protocols must enable the prover to convince the verifier that a given statement is true, while satisfying three properties: completeness, soundness, and zero-knowledge. *Id.*; *see also* Shafi Goldwasser et al., *The Knowledge Complexity of Interactive Proof Systems*, 18 SIAM J. COMPUTING 186, 189 (1989).
39. Feng & McMillin, *supra* note 38; *see also* Goldwasser et al., *supra* note 38, at 186–87.

important branch of cryptography and computational complexity theory with applications in cryptocurrencies,[40] smart contracts,[41] and warfare.[42]

A classic example of zero-knowledge proof involves a color-blind friend and two differently colored but otherwise identical balls.[43] In this example, the prover ("P") is trying to convince her color-blind friend, the verifier ("V"), that the two balls indeed are colored differently without revealing the balls' actual colors to V. The zero-knowledge protocol operates as follows: V initially presents one ball to P. V then puts the two balls behind her back, at which point she may or may not switch the two balls. V presents a ball to P a second time and asks whether V has switched the balls and is presenting a different ball from the first time. If the two balls are indeed of different colors, then P should be able to give the correct answer. If not, P will have to guess whether V switched the balls, with a 50% chance of guessing correctly. V then repeats this set of actions multiple times, so that P's chances of having arrived at the correct answer by guessing each time become infinitesimal. For example, if V repeats these actions twenty times, the chance of guessing it correctly is about one in a million — approximately the

40. *See, e.g.*, Yogita Khatri, *EY Reveals Zero-Knowledge Proof Privacy Solution for Ethereum*, COINDESK (Oct. 31, 2018), https://www.coindesk.com/ey-reveals-zero-knowledge-proof-privacy-solution-for-ethereum [https://perma.cc/CW7D-SFRX]; Matteo Campanelli et al., *Zero-Knowledge Contingent Payments Revisited: Attacks and Payments for Services*, 2017 ACM SIGSAC CONF. ON COMPUTER & COMM. SECURITY 229, 229.

41. *See, e.g.*, Ahmed Kosba et al., *Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts*, 2016 IEEE SYMP. ON SECURITY & PRIVACY 839, 839; Patrick McCorry et al., *A Smart Contract for Boardroom Voting with Maximum Voter Privacy*, 2017 INT'L CONF. ON FIN. CRYPTOGRAPHY & DATA SECURITY 357, 360.

42. *See, e.g.*, Sébastien Philippe et al., *A Physical Zero-Knowledge Object-Comparison System for Nuclear Warhead Verification*, 7 NATURE COMM. 12890, 12890 (2016).

43. This example was first demonstrated live by Kostas Chalkias and Mike Hearn at a conference in 2017. *See* Kostas Chalkias, *Demonstrate How Zero-Knowledge Proofs Work Without Using Maths*, LINKEDIN (Sept. 13, 2017), https://www.linkedin.com/pulse/demonstrate-how-zero-knowledge-proofs-work-without-using-chalkias [https://perma.cc/GJZ4- 7GV9].

same chance as getting struck by lightning in the US in any given year.[44] This example is illustrated in Figure 2 below.

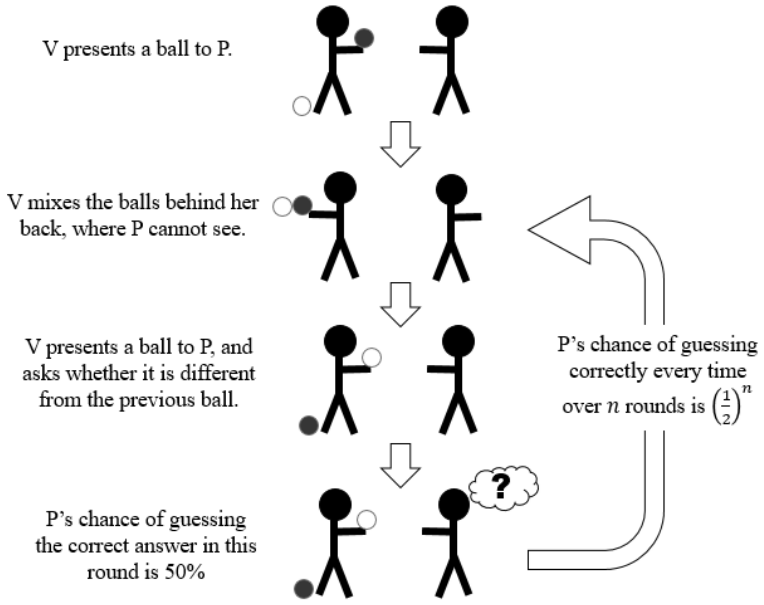

Figure 2: The Colored-Ball Example of Zero-Knowledge Proof

The example above illustrates how one zero-knowledge proof protocol works. The next two examples will be relevant to privilege log disputes, discussed in detail in Part IV. The first example involves two millionaires going to lunch together, who agree that the wealthier one between them should pay for the meal.[45] They want to figure out who is wealthier without revealing their actual wealth. Zero-knowledge proof can verify the truth of this statement without telling either millionaire about the content underlying the statement — i.e. their actual wealth. This can be achieved through a computer program that takes the input of wealth from both millionaires, compares the values, and outputs only the name of the wealthier millionaire. The fairness of the program could be guaranteed by making the algorithm open source and transparent. This zero-knowledge protocol is different from the color-blind friend example because it does not require repeated verification through probabilistic outcomes. Rather, the computer program

44. *How Dangerous is Lightning?*, NAT'L WEATHER SERV., https://www.weather.gov/safety/lightning-odds [https://perma.cc/8S8Y-FNH7].

45. Kroll et al., *supra* note 6, at 668.

knows the answer definitively (i.e. which person is wealthier) and simply withholds certain information (i.e. the wealth of each person).

The second example involves a Sudoku puzzle solved by the prover, who wishes to prove to a verifier that they have in fact solved the puzzle without revealing the actual solution.[46] Here, the statement to be proven is that the prover has solved the Sudoku puzzle. The content to be protected by the zero-knowledge protocol is the puzzle's actual solution. Under one possible protocol, the verifier will request a random row or column or square to be revealed. The zero-knowledge protocol then randomly permutes (i.e. randomly rearranges the sequence of) the numbers in the requested area such that there is a one-to-one mapping between the original numbers and the permuted numbers. See Figure 3 for an example of the permutation of a square of a Sudoku solution. The verifier can then check whether the row or column or square indeed contains nine unique numbers. The verifier can repeat these requests for each row, column, and square. As the number of requests increases to twenty-seven (the total number of rows, columns, and squares that contain nine unique numbers in Sudoku), the probability that the prover does not actually have the real solution decreases to zero. The verifier can then be satisfied with the veracity of the statement. Meanwhile, since the numbers change randomly every time, the verifier will not be able to piece together the actual solution to the Sudoku puzzle. Thus, the content behind the statement is protected.

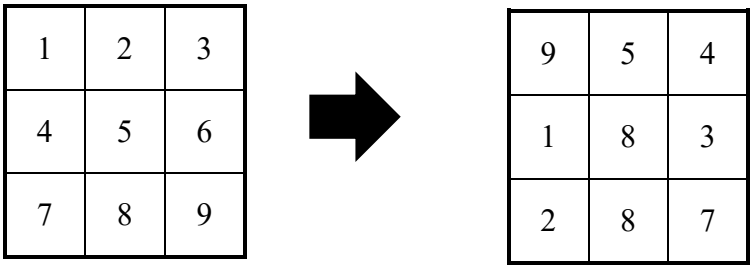

Figure 3: An Example of the Permutation of a Square of a Sudoku Solution

Both the color-blind friend and Sudoku protocol examples require interactions between provers and verifiers. These kinds of protocols are

46. *See* Manish Goregaokar, *Interactive Sudoku Zero-Knowledge Proof*, IN PURSUIT OF LAZINESS (Aug. 10, 2016), https://manishearth.github.io/blog/2016/08/10/interactive-sudoku-zero-knowledge-proof [https://perma.cc/JN4H-2LXJ].

called interactive zero-knowledge proofs.[47] In contrast, the Millionaire Model eliminates the interactions. This variant is called non-interactive zero-knowledge proof, which is often technical and involves extensive mathematics.[48]

In summary, zero-knowledge proof solves a trust problem: a prover wants to prove that she has a secret without telling a verifier what the secret is. This is the exact trust problem presented in privilege log disputes: the prover is the privilege-claiming party, and the verifier is the opposing party. The statement to be proven is that a certain document is indeed privileged. The prover desires to prove this statement without revealing the underlying content, i.e. the privileged documents themselves.

Zero-knowledge proof is fascinating for its seemingly contradictory definition in that it is "both convincing and yielding nothing except that the assertion is indeed valid."[49] This property has made zero-knowledge study an active research area.[50] It is not readily apparent whether anyone has contemplated the application of zero-knowledge proof in the context of discovery, especially privilege log disputes.

IV. SOLVING THE TRUST PROBLEM IN PRIVILEGE LOG DISPUTES WITH ZERO-KNOWLEDGE PROOF

Part IV proposes two zero-knowledge protocols to resolve privilege log disputes. The first protocol is modeled after the Millionaire Model discussed in Part III. It has two means of implementation which both involve machine learning algorithms. It is important to not confuse machine learning algorithms with zero-knowledge proofs. The first protocol below is still an application of zero-knowledge proof, which enables validation of a prover's secret without revealing it to the verifier. The protocol simply employs machine learning as a means to an end. The second protocol does not involve machine learning. It is modeled after the Sudoku example discussed in Part III.

A. The Millionaire Model

Drawing on existing zero-knowledge protocols, it would appear intuitive that the Millionaire Model[51] offers a solution: the privilege-claiming party submits the privileged documents in dispute to a zero-knowledge proof algorithm. Just as the algorithm in the Millionaire

47. *See* Goldwasser et al., *supra* note 38, at 189–91 (defining interactive proof systems).

48. To learn more about non-interactive zero-knowledge proof, see generally Manuel Blum et al., *Non-Interactive Zero Knowledge*, 20 SIAM J. COMPUT. 1084 (1991).

49. Feng & McMillin, *supra* note 38.

50. *Id.*

51. *See supra* Part III.

Model compares wealth and outputs only the wealthier person's name, the algorithm here can determine whether a document is indeed privileged and generates only a "yes" or "no" answer with a confidence level without revealing the underlying document.

Unlike the Millionaire Model, however, where the program only needs to compare two numerical values (a task computers are well-trained to do), discerning whether a document is privileged or not often requires machine learning.[52] This Note reviews the basics of these concepts and how they are used in current Technology-Assisted Review in e-discovery.

1. A Review of Machine Learning in Technology-Assisted Review in e-Discovery

A Technology-Assisted Review ("TAR") process involves human reviewers and computers collaboratively identifying documents in a collection relevant to a production request or identifying privileged documents to be withheld.[53] A typical protocol functions as follows: at the outset of the process, a human reviewer uses a keyword search to identify an initial set of documents to be reviewed and labeled as responsive or not ("coded").[54] These documents are commonly referred to as the "seed set," and are the initial inputs used to train a related learning algorithm. This algorithm scores each document in the collection by the likelihood that it is responsive. The human reviewer then reviews and codes the top-scoring documents. At this point, all documents that have been coded are collectively referred to as the "training set" to train the learning algorithm. The process of selecting the highest-scoring documents, reviewing and coding them, and adding them to the training set continues until "enough" responsive documents have been found.

52. *See* Maura R. Grossman & Gordon V. Cormack, *Technology-Assisted Review in E-Discovery Can Be More Effective and More Efficient than Exhaustive Manual Review*, 17 RICH. J.L. & TECH. 1, 4 (2011).

53. *Id.* at 3.

54. *See* Gordon V. Cormack & Maura R. Grossman, *Evaluation of Machine-Learning Protocols for Technology-Assisted Review in Electronic Discovery*, PROC. OF THE 37TH ANNUAL INT'L ACM SIGIR CONF. ON RES. & DEV. IN INFO. RETRIEVAL 153, 154 (2014) (describing the "CAL" protocol).

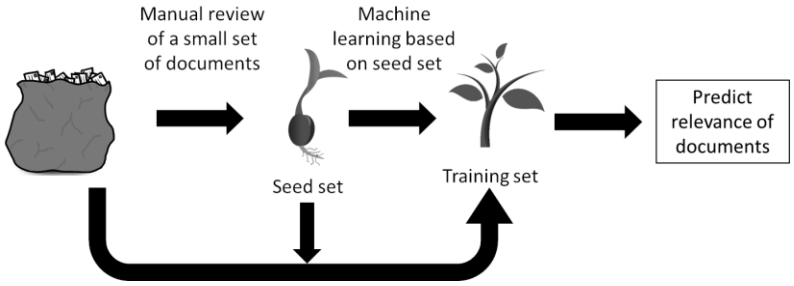

Figure 4: A Typical TAR Process to Identify Relevant Documents

What is "enough" is typically a legal determination governed by Federal Rules of Civil Procedure 26(g)(1) and 26(b)(1). The former requires an attorney of record to certify that "information, and belief formed after a reasonable inquiry" presented is based on her best knowledge.[55] The latter, in contrast, asks a court to limit discovery when "the burden or expense of the proposed discovery outweighs its likely benefit."[56] Taken together, the rules "require that discovery requests and responses be proportional."[57] Thus, what is "enough" depends on the burden associated with "find[ing] more responsive documents, and how important those documents would likely be in resolving the legal dispute."[58]

There are various approaches to selecting seed sets. These documents may be selected using keyword search, Boolean search, conceptual search, clustering, or sampling.[59] A keyword search uses keywords to retrieve specific documents "based on a priori knowledge of the search terms."[60] A Boolean search is a more powerful keyword search: it includes Boolean connectors such as AND, OR, and NOT, that aggregate keywords into more complex search phrases.[61] While keyword and Boolean searches play important roles in e-discovery, the reviewing attorneys do not always know the precise terms to formulate an effective search using simple terms, as they do not know the content of the privileged documents.[62] As a result, other tools are needed. Conceptual searches further enhance keyword and Boolean searches. They

55. FED. R. CIV. P. 26(g)(1).
56. FED. R. CIV. P. 26(b)(1).
57. Grossman & Cormack, *supra* note 52, at 6.
58. Cormack & Grossman, *supra* note 54, at 154.
59. Grossman & Cormack, *supra* note 52, at 4.
60. Shannon Brown, *Peeking Inside the Black Box: A Preliminary Survey of Technology Assisted Review (TAR) and Predictive Coding Algorithms for Ediscovery*, 21 SUFFOLK J. TRIAL & APP. ADVOC. 221, 255 (2016).
61. *Id.* at 258.
62. *Id.* at 259.

associate related keywords into conceptual groups.[63] For example, "check, bank, finance, and payment" is in a different conceptual group as "check, accountability, power, and encroachment." While both groups contain the word "check," the first group raises a different idea (a financial instrument) than the second (a stopping or slowing of progress). A conceptual search retrieves documents with language and ideas relevant to the keywords, rather than simply limiting the search to the keywords themselves.[64] Another approach to selecting seed documents is clustering. "Clustering tools use statistical methods to automatically group documents with similar content[s]" (e.g. concepts), for example by the number of words that overlap from one document to another.[65] The likelihood that two documents are related has a positive relationship with the number of words they share in common.[66] Finally, sampling (i.e. selecting documents at random) is yet another approach to construct seed sets.[67]

After seed documents have been selected, there are still many variations in training set expansion methods. One protocol, for example, begins with the creation of a seed set used to train a learning algorithm, but selects subsequent training documents to be reviewed and coded from documents that the learning algorithm is least certain about, as opposed to selecting top-scoring documents.[68] Reviewed documents are then added to the training set, and the process continues until the marginal benefit of including an additional document in the training set is outweighed by the marginal cost of reviewing and coding that document.[69] This is a point commonly referred to as "stabilization."[70] Stabilization is a desired goal in most machine learning protocols.

There are also many machine learning methods that a TAR programmer may choose from. Beyond clustering, described above,[71] logistic regression is another basic document classification method which estimates a document's relevance given its features (i.e. certain keywords).[72] This enables the algorithm to come up with a "best fit" from a training set and apply the fit to predict which documents are relevant.

---

63. *Id.* at 283.

64. *Id.* at 283–84.

65. Jacob Tingen, *Technologies-That-Must-Not-Be-Named: Understanding and Implementing Advanced Search Technologies in E-Discovery*, 19 RICH. J.L. & TECH. 1, 23 (2012).

66. *Id.*

67. Random sampling to generate seed sets has been shown to be statistically less effective than other techniques. *See* Cormack & Grossman, *supra* note 54, at 159.

68. *Id.* at 156.

69. *Id.* at 160.

70. *Id.* at 153.

71. *See* Tingen, *supra* note 65.

72. Maura R. Grossman & Gordon V. Cormack, *The Grossman-Cormack Glossary of Technology-Assisted Review*, 7 FED. CTS. L. REV. 85, 100 (2013). For technical details on logistic regression, see Brown, *supra* note 60, at 267–68 (describing that logistic regression classifies items with a sigmoid, step-function which assigns documents with values near either 0 or 1 to differentiate between classes of documents).

Finally, a third technique, called Support Vector Machines (SVMs), provides "powerful and reliable classifications."[73] This technique operates on the principle of a separating hyperplane with margins. In a simple example of classifying objects with only two features, one can think of a "hyperplane" as a line that linearly separates the two sets of objects. The further a data point is from the hyperplane, the more confident the algorithm is about the classification. An SVM thus "draws a line" based on the features analyzed in the training set to decide on which side of the hyperplane a new document lands, while considering the room for error provided by the margin.[74] Margins help provide a better overall predictive capacity.[75]

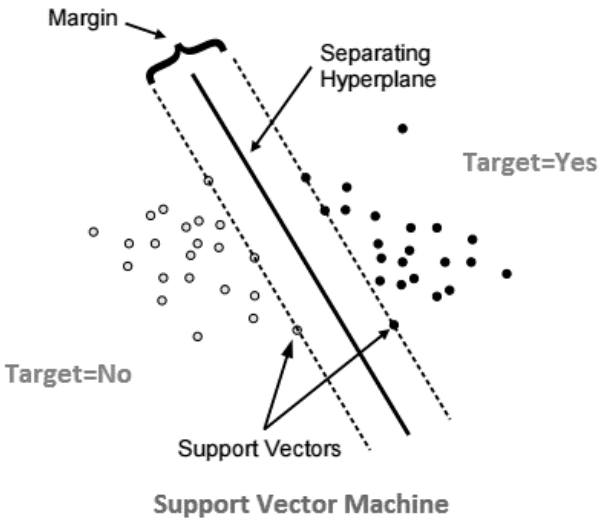

Figure 5: SVM Hyperplane with Margins[76]

In contrast to statistics-based tools such as clustering, logistical regression, and SVMs, Bayesian classifiers are based on probability algorithms that "determine th[e] likelihood that a document is relevant by placing a value on words, their relationships to each other, and their proximity and frequency in comparison with other documents."[77] Bayesian systems are informed by weighing and ranking words and

73. Brown, *supra* note 60, at 270.
74. *Id.* at 270–71.
75. *Id.* at 272.
76. *Building Predictive Model using SVM and R*, DNI INST. (Sept. 13, 2015), http://dni-institute.in/blogs/building-predictive-model-using-svm-and-r/ [https://perma.cc/8KX3-J5GC].
77. Tingen, *supra* note 65, at 25.

their relationships, and learn through the review process.[78] Since the late 1990s, email software has used Bayesian inferences to filter spam.[79] Spam filters compare incoming email against existing messages which have been flagged as junk or non-junk.[80] The proximity and frequency of pre-defined spam keywords (e.g., "Nigerian Prince," "wire transfer," and "bank routing number") within the incoming email are then compared against the existing messages.[81] The algorithm may also consider other parameters, such as the location of the key terms within the email, or whether the user has previously received a message from that email's sender.[82] In the TAR context, Bayesian technology requires a seed set of documents already sorted into privileged and non-privileged categories.[83] The software applies the categorizations of the seed set to the entire document collection, and verifies its privilege determinations through human input.[84] Over time, the algorithm can learn and categorize with high level of accuracy.[85]

Finally, there is natural language processing, which describes the entire "discipline that addresses fundamental issues of the computational processing of human languages."[86] Natural language processing "focuses on understanding language itself — including sentence parsing, word frequencies, . . . syntax, . . . word roots, and many other aspects of human language . . . ."[87] It enables software applications to use greater context to predict the most likely "meaning" of a sentence based on words, on paragraphs, or on the entire article, to extract their overall and contextual meanings.[88] This tool will be particularly helpful in TAR.

TAR is very effective. For some time, it was speculated that exhaustive manual review was more effective than TAR at finding the largest number of the most responsive documents.[89] Maura R. Grossman and Gordon V. Cormack debunked this myth by finding that TAR yields more accurate results than manual review with much lower effort.[90] Indeed, the *Merck* case discussed in Part II shows that even a

78. *Id.*
79. *Id.* at 26.
80. *Id.*
81. *Id.*
82. *Id.* at 27.
83. *Id.* at 28.
84. *Id.*
85. *Id.*
86. Brown, *supra* note 60, at 284.
87. *Id.* at 284–85. N-grams are one such implementation of natural language processing. It is "a generic name for a feature that typically associates two, three, four, or more words together as a single feature." *Id*. at 246.
88. *Id.* at 285.
89. *The Sedona Conference Best Practices Commentary on the Use of Search and Information Retrieval Methods in E-Discovery*, 8 SEDONA CONF. J. 189, 199 (2007).
90. Grossman & Cormack, *supra* note 52, at 43. In response to the article and the increasing adoption of TAR in legal proceedings, *see, e.g.*, Da Silva Moore v. Publicis Groupe, 287

diligent judge can be inconsistent and make mistakes when faced with the daunting task of going through a large volume of documents.[91] This finding increases the confidence that a zero-knowledge protocol is a more effective solution to privilege log disputes.

2. Case-Specific Machine Learning Algorithms

This Note returns to the discussion of how to use machine learning to implement a zero-knowledge protocol to resolve privilege log disputes. The idea is derived from the Millionaire Model where documents are fed into a machine learning algorithm, which then returns to the privilege-challenging party a yes/no response, and a confidence level.

Based on the review in the previous section, it appears that current TAR technology has the potential to reliably and accurately determine whether a document is privileged or not. However, this result can only be achieved through careful selection and human review of the initial seed set; and continuous training until stabilization is achieved. The documents used by most current TAR protocols to train algorithms are specific to each case — the keywords and concepts that machine learning algorithms rely on to make predictions are case-specific. For example, "solution," "concentration," and "weight" in close proximity to each other may well be a good indicator that a particular document qualifies for trade-secret privilege in a case involving intellectual property rights of a chemical company. The same words in an employment case for a software company probably would not be flagged as privileged.

Case-specific training presents a challenge to resolving privilege log disputes. Because the accuracy and reliability of a given machine learning algorithm depends on the method and quality of its training process, the only way for privilege-challenging parties to trust algorithm results is for that party to be involved in the training process. Indeed, in one of the first judicial opinions recognizing computer-assisted review as an acceptable way to search for relevant documents, *Da Silva Moore v. Publicis Groupe*,[92] the court blessed TAR because of the

F.R.D. 182 (S.D.N.Y. 2012), many legal scholars have begun discussions about counsels' legal obligation under Rule 26 when TAR is employed, *see, e.g.*, Karl Schieneman & Thomas C. Gricks III, *Implications of Rule 26(g) on the Use of Technology-Assisted Review with Foreward by John M. Facciola, U.S. Magistrate Judge*, 7 FED. CTS. L. REV. 247, 283 (2014) (stating "that counsel must know how each step . . . will impact the ultimate production so counsel will be able to meet the Rule 26(g) standards without fear of being sanctioned"); Kate Bauer, Leveling the Field: Playing Technology-Assisted Review by the [Federal] Rules (Oct. 1, 2018) (unpublished manuscript) https://ssrn.com/abstract=3279999 [https://perma.cc/9BT2-AWFX] (arguing "courts should embrace the unique opportunities TAR presents and address any associated discovery abuses using time-tested discovery rules").

91. *See* Vioxx Prods. Liab. Litig. Steering Comm. v. Merck & Co. (*Vioxx v. Merck*), No 06-30378, 06-30379, 2006 WL 1726675, at *2 (5th Cir. May 26, 2006).

92. 287 F.R.D. 182 (S.D.N.Y. 2012).

"transparency in the discovery process."[93] In that case, the court was faced with the question of whether to allow TAR to be used in the production of relevant documents. The plaintiffs supplied some of the keywords for the establishment of the seed set. The defendant agreed to turn over "all of the documents that are reviewed as a function of the seed set, whether they are ultimately coded as relevant or irrelevant, aside from privilege."[94] Documents coded as relevant and non-privileged would be reviewed by plaintiffs' counsel and, subject to their feedback, included in the seed set. The court noted that the defendant's transparency in the search protocol "made it easier for the Court to approve the use of predictive coding."[95]

Such transparency is challenging in the case of privilege log disputes for several reasons. First, it might be difficult for the document-requesting party to supply keywords or concepts to construct the initial seed set, as the documents are privileged and the requesting party does not know what is contained in those documents. Second, unlike in *Da Silva Moore*, due to the sensitive nature of privileged documents, allowing the requesting party in a privilege log dispute to participate in the construction of the seed set, or review or provide feedback to documents from sampling could create problems. Having the requesting party review these documents would practically defeat the purpose of zero-knowledge proof, which is to keep the producing party's documents secret from the requesting party.

The parties can still achieve transparency and maintain a zero-knowledge protocol if the privileged documents can only be reviewed by attorneys, and if inadvertently-disclosed privileged documents are subject to "clawback" agreements so that inadvertent disclosures do not constitute waivers. As the *Da Silva Moore* court astutely observed, "computer-assisted review is not a magic, Staples-Easy-Button, solution appropriate for all cases."[96] In cases where the privileged documents are so sensitive such that they cannot even be seen by attorneys of the opposing party, zero-knowledge proof protocol cannot be implemented through case-specific machine learning algorithms.[97] The natural next question is: can the algorithms be trained using generic documents?

93. *Id.* at 192.
94. *Id.* at 187.
95. *Id.* at 192.
96. *Id.* at 189.
97. Theoretically, a neutral arbitrator like a judge could conduct *in camera* review to participate in the construction of the seed set, or review or provide feedback to documents from sampling, but this would defeat the initial purpose of using zero-knowledge proof in resolution of privilege log disputes — to establish trust without an intermediary.

3. Generic Machine Learning Algorithms

A party may claim privilege for several reasons, and thus naturally classify privileged documents into several broad categories such as attorney-client privilege, work-product doctrine, marital privilege, religious privilege, physician-patient privilege, counseling and psychological privilege, and trade-secret privilege.[98] Most of these privileges have distinct patterns, such as being based upon the sender and recipient of communications, and whether privilege is asserted in the communications. However, these patterns are broad and can easily be abused if, for example, an attorney labels all her client communications as privileged.

Luckily, there are more clues that machine learning algorithms can potentially use in predictive coding. Each broad category of privilege may be further divided into subcategories. For example, attorney-client privilege in the context of corporate clients could have its own defining characteristics, which the Supreme Court outlined as protecting the communications of any employee who communicates with an attorney on behalf of the corporation if the communication concerns corporate matters within the scope of the employee's duties.[99] The features in this subcategory include attorney writing on behalf of the corporation, the recipient's employment status, and the recipient's work duties that might need manual input on a case-by-case basis.

Opinions of counsel regarding patent infringement is a distinct subcategory in the context of work-product privilege. When a party is concerned that it might be potentially infringing on another party's patent, it may obtain opinion letters from experienced attorneys opining that either the party is not infringing or that the patent-at-issue is invalid. Even if the party is later found to infringe, these letters may help the party avoid a finding of willful infringement and enhanced damages.[100] These opinion letters typically have set templates.[101] They start by discussing the legal standard and patent-at-issue, including term-by-term

98. FED. R. EVID. 501–02 advisory committee's note to 1974 enactment.

99. Upjohn Co. v. United States, 449 U.S. 383, 386 (1981).

100. In 1983, the Federal Circuit placed an affirmative duty on potential infringers to secure non-infringement and/or invalidity opinions of counsel to avoid findings of willful infringement and enhanced damages. Underwater Devices Inc. v. Morrison-Knudsen Co., 717 F.2d 1380, 1389–90 (Fed. Cir. 1983). The Federal Circuit subsequently changed course and adopted an "objective recklessness" standard that made willful infringement harder to prove. *In re* Seagate Tech., LLC, 497 F.3d 1360, 1371 (Fed. Cir. 2007). The Supreme Court overturned that precedent by adopting a "subjective recklessness" standard, focusing on the infringer's knowledge and belief as to infringement and validity at the time of infringement. Halo Elecs., Inc. v. Pulse Elecs., Inc., 136 S. Ct. 1923, 1933 (2016). This decision increased the relevance of obtaining opinions of counsel to show a defendant's state of mind at the time of infringement. *Id.*

101. *See, e.g.*, Lexis Practice Advisor, *Pre-Litigation Opinion Letter to Patent Owner (Patent Infringement and Validity)*, LEXISNEXIS (July 6, 2018), https://advance.lexis.com/api/

claim construction, drawing on specifications within the patent, file histories, and past court opinions.[102] A non-infringement letter would then turn to the activity of the potential infringer and identify the patent claim element(s) the activity does not practice to conclude that the activity does not infringe on the patent.[103] An invalidity opinion letter, in contrast, would review prior art and analyze how the patent claims are invalid in light of the prior art.[104]

Another subcategory of privileged documents with distinctive features common among most cases is the "invention record" protected by attorney-client privilege. [105] Specifically, "[i]nvention records are standard forms generally used by corporations as a means for inventors to disclose to the corporation's patent attorneys that an invention has been made and to initiate patent action."[106] Invention records are typically "short documents containing space for such information as names of inventors, description and scope of invention, closest prior art, first date of conception and disclosure to others, dates of publication, etc."[107] These distinct features, along with the sender and recipient of the documents, could make them identifiable by machine learning algorithms trained by these documents from a pool of generic cases.

As shown by the examples above, subcategories of privileged documents often have patterns or features defining them such that it is possible to train a machine learning algorithm with subcategory-specific generic documents instead of documents selected from the current case. Since the training documents are generic, the selection of training sets, the training process, and sampling all may be made transparent to all interested parties. The program itself should also be open source to maximize transparency and trustworthiness. This makes the program less susceptible to hackers who might game the system by feeding the program with engineered documents in an attempt to bias the prediction. Both parties involved in the privilege disputes can check the program for reliability.

Training machine learning algorithms for each subcategory of privileged documents may also benefit from unsupervised training. Human reviewers can provide documents from that subcategory and let the al-

---

permalink/e0aee9a4-7bfc-43cf-8b9b-0c964c521824/?context=1000522 (last visited May 10, 2019).

102. *See, e.g.*, *id.*

103. *See* Lexis Practice Advisor, *Pre-Litigation Opinion on Patent Infringement and Validity*, LEXISNEXIS (Nov. 16, 2018), https://advance.lexis.com/api/permalink/0a0d0701-df14-45c7-a6b1-62848b758ead/?context=1000522 (last visited May 10, 2019).

104. *See id.*

105. *See In re* Spalding Sports Worldwide, Inc., 203 F.3d 800, 806 (Fed. Cir. 2000) ("[T]he invention record of [a] patent is protected by the attorney-client privilege . . . .").

106. *Id.* at 802 n.2.

107. *Id.*

gorithm identify patterns in the documents without further human intervention. Unsupervised training excels at detecting latent structure or density patterns in data, and may identify features that elude human reviewers.[108]

B. The Sudoku Model

The Sudoku zero-knowledge protocol works by scrambling the numbers in a row, column, or square each time a verifier requests a check.[109] This method could potentially be used in a zero-knowledge protocol resolving privilege log disputes. If TAR is used during the e-discovery process, a document will be identified as likely-privileged based on keywords, conceptually related words, and their proximity to each other.[110] It is possible to scramble (i.e. to substitute the original words with others) or mask these words such that the document retains a format recognizable as privileged to the requesting party while withholding the sensitive underlying content. This approach might be particularly effective for documents covered by trade-secret privilege, where scrambling certain numbers, ranges, or even orders of steps may prevent others from learning the trade secret.

To illustrate how this approach works, consider the following hypothetical Coca-Cola recipe,[111] which is one of the most popular examples of trade secrets:

Ingredients:
- 1 oz citrate caffeine
- 30 lbs. sugar
- 3 oz citric acid
- 4 oz F.E. coco
- 1 oz ext. vanilla
- 2.5 gal. water
- 1 qt. lime juice
- caramel – sufficient
- 2.5 oz flavorings

Flavorings (7X formula):
- 80 oil orange
- 40 oil cinnamon
- 120 oil lemon
- 20 oil coriander
- 40 oil nutmeg
- 40 oil neroli
- 1 qt. alcohol

Directions:
- Mix caffeine, acid and lime juice 1qt.
- Boiling water add vanilla and flavorings when cool.
- Let stand for 24 hours.

Figure 6: A Hypothetical Coca-Cola Recipe

108. Brown, *supra* note 60, at 264–65.

109. *See supra* Part III for a detailed description of the Sudoku Model.

110. *See supra* Part IV.A.1 for a review of TAR.

111. Richard Grove, Notes on Making Cola 12 (Aug. 2005), https://sparror.cubecinema.com/cube/cola/chemistry/cola.pdf [https://perma.cc/VMD8-K8MZ].

One way to "scramble" this trade secret is to substitute original numbers, units, and ingredients with ones randomly generated by an algorithm. The number of ingredients and order of directions may also change. The resulting recipe may appear as illustrated in Figure 7.

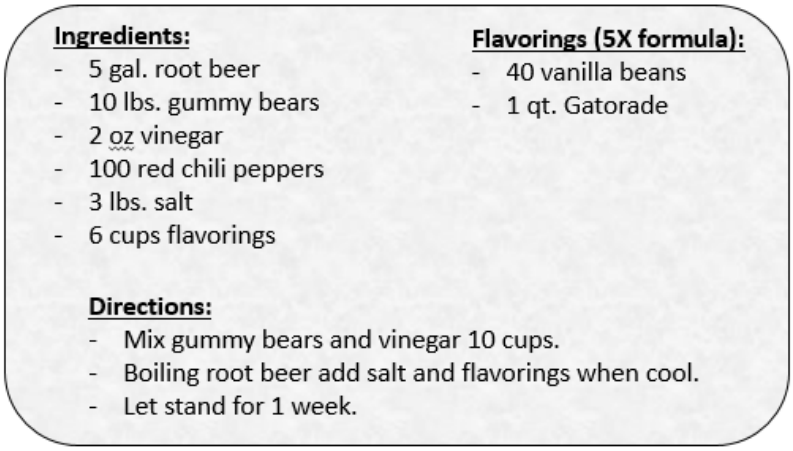

Figure 7: Altered Hypothetical Coca-Cola Trade Secret Pursuant to the Sudoku Zero-Knowledge Protocol

It is difficult to predict whether or not this approach would still leave enough information for the requesting party to determine the original content in the privileged document. It is possible that the privilege-claiming party would not even want to reveal the basic format of the document in dispute. In patent litigation, for example, an alleged patent infringer sometimes will seek and obtain opinions of counsel that it does not infringe. These opinion letters are typically protected by attorney-client privilege or work-product doctrine. Depending on the litigation strategy, the alleged infringer may not choose to assert defenses based on these opinion letters (especially if they contain shaky legal opinions).[112] As a result, the alleged infringer may not want to disclose the existence of the letters to their opponent. By revealing the format of the documents, the opponent may not learn about the content of the opinion letters, but it would know they exist. While this may not be desirable, the scrambling approach of the zero-knowledge protocol at least does not perform worse than the status quo, because the description required for privilege logs likely would give away the existence of those opinion letters. The effectiveness of this approach will likely require further investigation.

112. *See, e.g.*, Cobalt Boats, LLC v. Brunswick Corp., 296 F. Supp. 3d 791, 802 (E.D. Va. 2017) (illustrating a situation in which the defendant refused to assert advice of counsel and waive the associated privilege).

V. CONCLUSION

E-discovery has significantly increased the number of documents to be reviewed for production. This in turn has increased the number of documents to be recorded on privilege logs, which has led to more privilege log disputes. At the core of these disputes is the tension between proving to the requesting party that the documents are truly privileged and the need to maintain the documents' confidentiality. Zero-knowledge proof is designed to allow an honest verifier to determine the veracity of a prover's statement without learning any other information.

One possible zero-knowledge protocol to resolve privilege log disputes is to input the document-in-question into a machine learning system that has been trained to distinguish privileged from non-privileged documents, and have the algorithm output only the answer and its confidence level to the requesting party. The algorithm can be trained in two proposed ways. The first training method uses case-specific documents as seed and training sets to predict whether another document in the collection is privileged. To ensure transparency, the requesting party must be able to participate in the creation of the seed set and provide feedback. This might require designating certain privileged documents as "for attorneys' eyes only" and instituting robust clawback provisions. The second training method does not have that limitation. Under this method, the algorithm is trained to identify specific types of privileged documents by learning from generic documents of that type.

It is also possible to construct a zero-knowledge protocol without machine learning by masking the keywords, concepts, numbers, and order of paragraphs such that the content cannot be guessed while retaining the form of the document to be recognizable as privileged. This Note identifies an opportunity for the application of zero-knowledge proof in privilege log disputes. There are many zero-knowledge protocols in active research and studies, and there are many opportunities to use them to make resolution of privilege log disputes simpler and more cost-effective.