Evaluating Source Code Disclosure in the Criminal Justice System

Greg Schwartz

Table of Contents

I. Introduction

The criminal justice system is becoming increasingly automated. Predictive algorithms are being employed for pre-trial investigations,[1] assessments of evidence,[2] sentencing,[3] and parole.[4] In response, a large body of work has developed on the fairness[5] and transparency[6] of these systems, but only recently has the academic discussion addressed the role of ownership and trade secrets.[7] The Sixth Amendment contains the right of a defendant to confront "the witnesses against him,"[8] which the Supreme Court has recognized with full force in regard to forensic evidence.[9] Yet in court, the developers of these criminal justice software systems often claim that the source code for their products are trade secrets and exempt from discovery. They challenge subpoenas from defendants and resist disclosure. They argue that revealing their code, even to just defense experts, would harm themselves and not provide any uniquely valuable information.[10]

---

[1] Bennett Moses, Lyria, and Janet Chan. *Algorithmic prediction in policing: assumptions, evaluation, and accountability*. Policing and society 28.7 (2018): 806-822. and Asher, Jeff, and Rob Arthur. *Inside the algorithm that tries to predict gun violence in Chicago*. The New York Times 13 (2017).
[2] Friedman, Jennifer, and Jessica Brand. *It Is Now up to the Courts: Forensic Science in Criminal Courts: Ensuring Scientific Validity of Feature-Comparison Methods*. Santa Clara L. Rev. 57 (2017): 367.
[3] Wexler, Rebecca. *Code of Silence: How private companies hide flaws in the software that governments use to decide who goes to prison and who gets out*. (2017).
[4] Wexler, Rebecca. *When a computer program keeps you in jail: How computers are harming criminal justice*. New York Times 13 (2017).
[5] Angwin, Julia & Larson, Jeff. *Bias in Criminal Risk Scores Is Mathematically Inevitable, Researchers Say*. ProPublica (identifying concerns over accuracy, objectivity, errors, and bias).
[6] Skeem, Jennifer. *Scientific Risk Assessment in Sentencing May Beat the Alternative*. Berkeley Blog; in contrast to Rebecca Wexler, *Code of Silence supra* note 3.
[7] Wexler, Rebecca. *Life, liberty, and trade secrets: Intellectual property in the criminal justice system*. Stan. L. Rev. 70 (2018): 1343.
[8] U.S. Const. amend. VI.; Crane v. Kentucky, 476 U.S. 683, 686–87 (1986); *see* United States v. Bess, 75 M.J. 70, 75–76 (C.A.A.F. Jan. 6, 2016)
[9] *Melendez-Diaz* 557 U.S. at 313
[10] *See Computers Are Helping Justice*, CYBERGENETICS (June 16, 2017), https://perma.cc/XNW3-Q4A6. The company has presented similar arguments in court. *See, e.g.*, State v. Fair, No. 10-1-09274-5 SEA (Wash. Super. Ct. King Cty. Apr. 1, 2016).

One such challenge was made in a federal court in Texas, where the company TLO argued that disclosing information about their Child Protection System (CPS) would "significantly degrade the usefulness of CPS as a worldwide investigative tool, as well as compromis[e] numerous ongoing criminal investigations."[11] While CPS offers a broad suite of tools, the component allegedly at risk of being compromised is its hashing algorithm. CPS is able to identify for law enforcement IP addresses that download "notable" images and videos.[12] Media is considered notable if it matches the hashes of files previously labeled by law enforcement as containing child pornography. TLO argues that disclosing information about their hashing algorithm will allow criminals to evade detection.

This is not the only time that defendants have been refused information on hashing systems. In New Mexico, Microsoft argued against disclosing their PhotoDNA algorithm and related software.[13] A Federal Court last year denied the defendant access to a report generated by the hashing algorithm used by Google to search images on Gmail.[14] This year, Apple nearly joined the fray with their own CSAM hashing technology.[15] While the Apple's system was recalled due to public outcry, the substantial technological effort demonstrates the resources being put into developing more hashing systems. With this trend comes growing need for a technological evaluation of these developers' claims that disclosure will compromise their products.

---

[11] U.S. v. Ocasio 2013 WL 2458617
[12] *See Inside the surveillance software tracking child porn offenders across the globe"*https://www.nbcnews.com/tech/internet/inside-surveillance-software-tracking-child-porn-offenders-across-globe-n1234019
[13] United States v. Rosenschein 2020 WL 3572662
[14] United States v. Miller 982 F.3d 412
[15] *See CSAM Detection Technical Summary* https://www.apple.com/child-safety/pdf/CSAM_Detection_Technical_Summary.pdf

The fight for disclosure reaches far beyond hashing algorithms. A particularly high-profile example is COMPAS, one of the most popular recidivism predictors nationwide.[16] Despite – or, perhaps, because of – academic and media criticism of its product,[17] the developer of COMPAS maintains that the weight of the COMPAS inputs for should be protected under trade secrecy.[18] In a similar vein, Cybergenetics argues that courtroom disclosure of its product TrueAllele would be "financially devastating."[19] These cases differ from the hashing algorithms because their primary concern is not giving criminals an advantage, but competitors. As these products continue to be more and more widely used, further analysis of the risk that disclosure poses to these companies is necessary as well.

In response to these growing issues, this paper will attempt to analyze the benefits and risks of disclosure. In Part II, it will provide context for the role and reception of software in the United States Criminal Justice System. In Part III, it will explore the benefits of disclosure by comparing the evaluation of software with and without access to source code. In Part IV, it will examine the arguments against disclosure that developers have argued in court. Finally, in Part V it will conclude that increased source code disclosure seems to be warranted.

Ultimately, this paper aims to provide an initial foray into the broad and under-analyzed technical claims being made by developers in criminal proceedings. By critically examining the

---

[16] *How We Analyzed the COMPAS Recidivism Algorithm* https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm

[17] Yong, Ed. *A popular algorithm is no better at predicting crimes than random people*. The Atlantic 17 (2018); *see also Layers of bias: A unified approach for understanding problems with risk assessment*. Criminal Justice and Behavior 46.2 (2019): 185-209.

[18] State v. Loomis 371 Wis.2d 235

[19] *Supra* note 3; *see also When a Computer Program Keeps You in Jail* https://www.nytimes.com/2017/06/13/opinion/how-computers-are-harming-criminal-justice.html (TrueAllele has submitted affidavits across the country alleging that disclosing source code to defense attorneys would cause "irreparable harm.")

costs and benefits of disclosure, this paper aims to help create a foundation for future research and better equip courts to adjudicate between developers and defendants.

II. Automation and Criminal Justice

A. The Rise of Automation

In recent years the United States government has delegated many of its functions to automated decision making.[20] But preceding this rise of automation, came a proliferation of expert testimony. A study of over 500 American trials in 1985 and 1986 found that expert testimony was used in eighty-six percent of trials. One commenter noted that in the United States, trial by jury was evolving into trial by expert, with experts "readily availab[le]" to testify in favor of each party.[21]

This increased use of experts comes with a heavily reliance on their expertise, but automation has narrowed the scope of expert authority considerably. Experts used to take the stand to provide testimony on experiments that they had personally conducted by hand, such as a forensic scientist describing their fingerprint identification procedure. Today, these expert witnesses tend to speak to an automated process that they oversaw, testifying the results of DNA mixture analysis software like TrueAllele with no knowledge of the underlying source code. This is now affects the vast majority of DNA testimony, as eighty-five percent of American DNA laboratories are using automated techniques.[22]

Given the shrinking expertise of the witnesses conducting forensic tests, it would seem reasonable for courts to implement separate assurances regarding the robustness and validity of

[20] Danielle Keats Citron, *Open Code Governance*, 2008 U. CHI. LEGAL F. 355, 356-57
[21] William T. Pizzi, *Expert Testimony in the US*, 145 NEW L.J. 82, 82 (1995)
[22] William C. Thompson and Dan E. Krane, *DNA in the Courtroom*, in PSYCHOLOGICAL & SCIENTIFIC EVIDENCE IN CRIMINAL TRIALS § 11:23 (2003).

the software, or at least to allow defendants to perform an examination. Such inspections are commonplace in civil litigation, even though the litigants are typically business competitors. It is so common in fact, that in the United States it is an aberration for a court to refuse requests from civil litigants challenging the reliability of software to have an independent expert analyze the code.[23] However, in criminal settings the source code generally has only circumstantial relevance, given that reviewing the code functions to validate the scientific equipment which generates direct evidence.[24]

B. Evidence Rules for Scientific Evidence

With this rise in expert testimony and automation have come new standards in the courtroom. These standards generally apply to two situations: admissibility and pre-trial discovery.

1. Admissibility

Until the 1970s, the federal courts and majority of state courts used the *Frye* standard of general acceptance for the admissibility of scientific evidence.[25] The court recognized that evidence must have "gained general acceptance" from the scientific community in order to be sufficiently established. This general acceptance would bring the evidence from "experimental" to "demonstrable," making it acceptable to use in the courtroom. Since general acceptance by the scientific community was necessary, one or even several expert opinions may not have bene sufficient to establish evidence as admissible.[26]

---

[23] Christopher M. Mislow, *Protecting Source Code from Disclosure During Pretrial Discovery*, 12 UTAH B.J. 39, 47 (1984).
[24] Edward J. Imwinkelried, *Computer Source Code: A Source of the Growing Controversy Over the Reliability of Automated Forensic Techniques*, 66 DEPAUL L. REV. 97 (2016).
[25] Frye v. United States, 293 F. 1013, 1014 (D.C. Cir. 1923)
[26] See Paul C. Giannelli, The Admissibility of Novel Scientific Evidence: Frye v. United States, A Half-Century Later, 80 COLUM. L. REV. 1197, 1204 n.41 (1980);

This bright line changed in 1975, when Article VII of the Federal Rules of Evidence took effect.[27] This Article removed the "general acceptance" language and replaced it with what would be interpreted as Daubert's Five-Factor Balancing Test.[28] While the checklist is flexible,[29] the *Daubert* definitively shifted the "gatekeeping role" from the scientific community to the judge.[30]

Proof of validation remains an essential under this Daubert standard, although courts have found that developers can meet this burden without offering testimony about the validity of the source code. Instead, as prosecutors have successfully argued, the government can meet that burden by presenting testimony about validation studies investigating the accuracy of the software.[31]

2. Pre-trial Discovery

While prosecutors have almost uniformly prevailed under admissibility standards, pre-trail discovery has been more mixed.[32] There have been two waves of source code discovery cases, the first regarding breath testing devices and the second regarding probabilistic genotyping programs. While there were some exceptions,[33] in the first wave the courts almost uniformly

[27] Act of Jan. 2, 1975, Pub. L. No. 93-595, 88 Stat. 1926 (codified at 28 U.S.C. §§ 2072, 2075 (2012))
[28] Katherine L. Moss, Note, The Admissibility of TrueAllele: A Computerized DNA Interpretation System, 72 WASH. & LEE L. REV. 1033, 1060–62 (2015).
[29] Daubert v. Merrell Dow Pharms., Inc., 509 U.S. 579, 588 (1993).
[30] *Id.* at 594
[31] Joe Palazzolo, *Judge Denies Access to Source Code for DNA Software Used in Criminal Cases*, WALL ST. J. (Feb. 5, 2016, 10:56 AM), http://blogs.wsj.com/law/2016/02/05/judge-deniesaccess-to-source-code-for-dna-software-used-in-criminal-cases/
[32] *Supra* note 24
[33] State v. Bjorkland, 924 So. 2d 971, 973 (Fla. Dist. Ct. App. 2006); State v. Underdahl, 767 N.W.2d 677, 683–84 (Minn. 2009); In re Comm'r of Public Safety, 735 N.W.2d 706, 712–13 (Minn. 2007); State v. Chun, 923 A.2d 226, 226–27 (N.J. 2007), cert. denied, 555 U.S. 825 (2008).

rejected requests for defense experts to be given access to source code.[34] The second wave, while ongoing, appears to be even more lopsided.[35]

Similarly to the reasoning in regard to admissibility, these courts reject defendant requests because they consider source code access to be unnecessary for validating the accuracy of software.[36] In addition, they reject discovery requests on the grounds that disclosing source code risks the financial wellbeing of the developers,[37] and the effectiveness of their products.[38]

III. The Importance of Source Code

However, it appears that knowledge of the underlying source code is necessary to establish an adequate understanding of a program's behavior and output. This is determined by first examining the proliferation of software errors, many of which are only caught through an examination of source code, and then by evaluating the alternative – peer-viewed black box validation studies. Finally, special attention is given to the specific problems with validation studies for DNA Mixture Analysis, the most common source of code disclosure cases.

A. Software Errors

Misbehavior in software is common, even in code developed and verified by leading experts. An infamous example comes from the hole in the ozone layer that went unnoticed for

[34] Moe v. State, 944 So. 2d 1096, 1097 (Fla. Dist. Ct. App. 2006); State v. Underdahl, 749 N.W.2d 117, 120–21 (Minn. Ct. App. 2008), aff'd in part, rev'd in part, 767 N.W.2d 677 (Minn. 2009) (affirming the district court's denial of production of computer code); State v. Burnell, No. MV06479034S, 2007 WL 241230, at *2 (Conn. Super. Ct. Jan. 18, 2007); State v. Walters, No. DBDMV0503409978, 2006 WL 785393, at *1 (Conn. Super. Ct. Feb. 15, 2006); People v. Cialino, 831 N.Y.S.2d 680, 681–82 (Crim. Ct. 2007).
[35] Commonwealth v. Robinson, No. CC 201307777 (Pa. Ct. C.P. Allegeheny Cty. Feb. 4, 2016) (discussing the admissibility of test results without disclosing source code in previous trials).
[36] Commonwealth v. Foley, 38 A.3d 882 (Pa. Super. Ct. 2012) (The court ruled that the available validation studies established the reliability of TrueAllele); *see also* Commonwealth v. Robinson, No. CC 201307777 (Pa. Ct. C.P. Allegeheny Cty. Feb. 4, 2016)
[37] *Supra note 18*
[38] *Supra* note 11, 13, 14

years because NASA programmers set their software to ignore "unrealistic" outlier data.[39] More recently, a software error from a misplaced less-than symbol (<) in Ireland's National Integrated Medical Imaging System caused "potentially thousands of patient records from MRIs, X-rays, CT scans, and ultrasounds" to be recorded incorrectly.[40] A software error caused a large Australian bank misreport certain transactions for almost three years, leading to widespread money laundering.[41] In rare cases, software is intentionally deceptive, as when Volkswagen programmed its vehicles to cheat at emissions tests.[42]

In short, errors are inevitable. The errors in these examples come from software built by large, trusted entities and responsible for important, sensitive information. As software will continue to be subject to bugs, and as programs become more complex, errors are liable to become even more widespread.[43]

Forensic software is not immune to these problems, and access to source code has repeatedly uncovered defects. We can see this in the Forensic Statistical Tool, a probabilistic genotyping program developed in 2010 by New York City's Office of the Chief Medical Examiner (OCME). Despite being a public entity, OCME spent years fighting any independent review of FST's source, even under a protective order. This finally ended in 2016, during a

---

[39] Michael King & David Herring, *Research Satellites for Atmospheric Sciences, 1978-Present, Serendipity and Stratospheric Ozone* (Dec. 10, 2001), https://earthobservatory.nasa.gov/Features/ RemoteSensingAtmosphere/remote _sensing5.php.

[40] Jack Power, *Software Company Behind HSE Scan Glitch Begins Investigation*, IRISH TIMES (Aug. 5, 2017), https://www.irishtimes.com/news/ireland/irish-news/software-company-behind-hse-scan-glitch-begins-investigation-1.3178349

[41] Allie Coyne, CBA Blames Coding Error for Alleged Money Laundering, itnews (Aug. 7, 2017), https://www. itnews. com.au/news/ cba-blames-coding-error-for-alleged -money-laundering-470233.

[42] Guilbert Gates et al., *Explaining Volkswagen's Emissions Scandal*, N.Y. TIMES (July 19, 2016), http://www.nytimes.com/interactive/2015/business/international/vw-diesel-emissionsscandalexplained.html?_r=0; Russell Hotten, Volkswagen: The Scandal Explained, BBC (Dec. 10, 2015), http://www.bbc.com/news/business-34324772; Sonari Glinton, How a Little Lab in West Virginia Caught Volkswagen's Big Cheat, NPR Morning Edition (Sept. 24, 2015), http://www.npr.org/2015/09/24/44305367 2/how-a-little-lab-in-west-virginia-caught -volkswagens-big-cheat

[43] Roger A. Grimes, *Five Reasons Why Software Bugs Still Plague Us*, CSO Online (July 8, 2014), https://www.csoonline.com/ article/2608330/security/5-reasons-why -software-bugs-still-plague-us.html.

criminal case when a federal judge ordered OCME to turn over FST's source code for review. [44] Over the process of that review, the defense expert identified many critical issues. There was a "secret function" that "tending to overestimate the likelihood of guilt." [45] The software did not use the methodology described in sworn testimony and peer-reviewed publications.[46]

Following these findings, STRmix – the software chosen to replace FST in New York – was analyzed by independent researchers and found to have programming errors that created false results in 60 out of 4,500 cases in Queensland, Australia.[47]

In a wave of cases involving breath testing devices for drunk driving, courts repeatedly asserted that there was no need for discovery of the source code because the defense had "other avenues of challenge."[48] They noted that defendants could access the device's calibration records and the operator's checklist in order to determine whether the device was working and properly used.[49] Yet, an eventual study of the source code for New Jersey's breath testing devices "uncovered a variety of defects that could impact the test results."[50] The courts had ignored the fact that checks for human error cannot identify software defects, and defendants paid the price.

---

[44] See United States v.Johnson, 15-CR565 (VEC) (S.D.N.Y.June 7, 2016).
[45] Stephanie J. Lacambra, Jeanna Matthews, and Kit Walsh. *Opening the Black Box: Defendants' Rights to Confront Forensic Software* https://lin-web.clarkson.edu/~jmatthew/publications/FINALARTICLE_pg28-39_Lacambra_Forensic_Software_May_2018_07102018_608BCX.pdf
[46] *Id.*
[47] David Murray, *Queensland Authorities Confirm 'Miscode' Affects DNA Evidence in Criminal Cases*, COURIER MAIL (Mar. 20, 2015), http://www.couriermail.com.au/news/ queensland/queensland-authorities -confirm-miscode-affects-dna-evidence-in -criminal-cases/news-story/833c580d3f1c 59039efd1a2ef55af92b.
[48] People v. Robinson, 860 N.Y.S.2d 159, 165 (App. Div. 2008) (quoting People v. Alvarez, 515 N.E.2d 898, 900 (N.Y. 1987)).
[49] Robinson, 860 N.Y.S.2d at 166 (discussing "calibration records," records which are "showing "that the machine was . . . properly maintained or that the test was . . . properly administered").
[50] State v. Underdahl, 767 N.W.2d 677, 685 (Minn. 2009) (describing a report in State v. Chun, 943 A.2d 114, 132–33 (N.J. 2008)).

B. The Sufficiency of Validation Studies

The courts have repeatedly accepted validation studies to demonstrate the performance of forensic software.[51] In the case of TrueAllele, courts have decided that the existence of validation studies makes source code disclosure unnecessary.[52] However, considering the proliferation of coding errors, it is unclear what guarantees these validation studies are able to provide.

The errors described in the above section create distinction between software and scientific principles that has not been acknowledged in legal scholarship. The procedures tested by empirical validation studies in chemistry or epidemiology rely on universal scientific principles. These principles motivate the reaction that one chemical has when exposed to another. And while software is subject to coding errors, these scientific principles have no such bugs. In "The Problem With Software and Its Assurance" John Rushby writes

> "the traditional disciplines are founded on science and mathematics and are able to model and predict the characteristics and properties of their designs quite accurately, whereas software engineering is more of a craft activity, based on trial and error rather than calculation and prediction."[53]

---

[51] Joe Palazzolo, *Judge Denies Access to Source Code for DNA Software Used in Criminal Cases*, WALL ST. J. http://blogs.wsj.com/law/2016/02/05/judge-deniesaccess-to-source-code-for-dna-software-used-in-criminal-cases/; *see* DAVID KAYE ET AL., THE NEW WIGMORE, A TREATISE ON EVIDENCE: EXPERT EVIDENCE 263–77 (2016 Supp.); Katherine L. Moss, Note, The Admissibility of TrueAllele: A Computerized DNA Interpretation System, 72 WASH. & LEE L. REV. 1033, 1060–62 (2015). Moss addresses the following decisions, all of which that the available validation studies established the reliability of TrueAllele: Queen v. Colin Duffy & Brian Shivers, [2011] NICC 37 (N. Ir. Crim.).; Commonwealth v. Foley, 38 A.3d 882 (Pa. Super. Ct. 2012).; Ohio v. Shaw, CR-13-575691, at *21 (Cuyahoga Ct. C.P. Oct. 10, 2014).

[52] Paula Reed Ward, *Judge to Allow DNA Evidence in Deaths of Wolfe Sisters*, PITT. POSTGAZETTE (Oct. 30, 2015, 12:00 AM), http://www.post-gazette.com/local/city/2015/10/30/allowDNA-evidence-in-deaths-of-East-Liberty-Wolfe-sisters/stories/201510300182; *see* Susan A. Greenspoon et al., *Establishing the Limits of TrueAllele Casework: A Validation Study*, 60 J. FORENSIC SCI. 1263 (2015); Mark W. Perlin et al., *TrueAllele Casework on Virginia DNA Mixture Evidence: Computer and Manual Interpretation in 72 Reported Criminal Cases*, 9 PLOS ONE (2014); Mark W. Perlin et al., *New York State TrueAllele Casework Validation Study*, 58 J. FORENSIC SCI. 1458 (2013)

[53] John Rushby. *Formal methods and their role in the certification of critical systems*. Safety and reliability of software based systems. Springer, London, 1997. 1-42.

Certainly chemical and epidemiological procedures are subject to human error as well, but only in software do we find widespread, elusive, systematic human error built into the innerworkings of the operation.

There are many reasons then why validation studies cannot always be counted on to catch software errors. Validation studies do not test every possible circumstance that may cause a bug to arise. Depending on the kind of error, it may not be obvious when an error has occurred. But beyond the theoretical explanations, as we saw in the section above, history shows that software errors exist despite validation studies.

These challenges for validation, it is useful to examine practices within the Computer Science discipline. To start, experiments are generally understood differently. Experiments are frequently used as proofs of concept rather than validation for a specific product.[54] In cases where specific products are being tested, the limits of empirical tests are widely recognized. For life-critical software, scholars have found that experimental validation is insufficient.[55]

Experimental validation is insufficient when the tests are not broad enough to capture undesirable behavior. To supplement these experiments, best practices in Computer Science include adherence to industry standards and other formal methods.[56] Still, even with these best practices, "the gap between the dependability requirements and what we can achieve in terms of delivering and measuring such dependability is huge."[57]

---

[54] *See* Marvin Zelkowitz and Dolores Wallace, *Experimental models for validating technology*. Computer 31.5 (1998): 23-31. ("Experimentation is one of those terms frequently used incorrectly in the computer science community… Here, 'experiment' really means an example that the technology exists or an existence proof that the technique can be employed. Very rarely does it involve any collection of data to show that the technology adheres to some underlying model or theory of software development, or that it is effective")

[55] Ricky W. Butler George B. Finelli. *The infeasibility of experimental quantification of life-critical software reliability*. Proceedings of the conference on Software for critical systems. 1991.

[56] Bowen, Jonathan, and Victoria Stavridou. *Safety-critical systems, formal methods and standards*. Software engineering journal 8.4 (1993): 189-209.

[57] Bowen, Jonathan, and Victoria Stavridou. *Formal methods and software safety*. Safety of Computer Control Systems 1992 (SAFECOMP'92). Pergamon, 1992. 93-98.

This is a problem faced by manufacturers of safety-critical systems who must release products quickly to remain competitive, while also ensuring they are safe for public deployment. Interesting work has been done on for manufacturers can navigate this limited-information problem, with some researchers using Bayesian estimates of the bug discovery rate[58] and defining manufacturer's position as a belief-state Markov decision process.[59] But crucially, these models assume that existing bugs are identified,[60] something that was rarely seen in the above section when source code remained secret.

While creating industry standards and other formal methods could make an attractive alternative to disclosing source code, the success of these processes relies on their actual adoption. Without source code disclosure, it is unclear if errors would ever surface, which compromises incentives for developers and the potential for formal methods as a solution. For these reasons, the scholarship ultimately seems to suggest that reviews of the source code are essential for discovering errors and truly validating software.

C. DNA Mixture Analysis

Developers of DNA mixture analysis software frequently argue that access to source code is unnecessary because validation studies in peer-reviewed journals are sufficient to establish efficacy.[61] Cybergenetics is the particularly notorious for this, as it refuses to make its TrueAllele source code available for inspection to any third party. The courts' acceptance of this refusal is

---

[58] Tom Chavez. *A decision-analytic stopping rule for validation of commercial software systems*, IEEE Transactions on Software Engineering, vol. 26, no. 9, pp. 907– 918, 2000.
[59] Jeremy Morton, Tim A. Wheeler, and Mykel J. Kochenderfer. *Closed-loop policies for operational tests of safety-critical systems*. IEEE Transactions on Intelligent Vehicles 3.3 (2018): 317-328.
[60] *Id*.
[61] *See Computers Are Helping Justice*, CYBERGENETICS (June 16, 2017), https://perma.cc/XNW3-Q4A6. The company has presented similar arguments in court. *See, e.g.*, State v. Fair, No. 10-1-09274-5 SEA (Wash. Super. Ct. King Cty. Apr. 1, 2016).

troubling since – beyond the general risk of missing software errors – DNA mixture analysis validation is particularly limited.

Compared to other software tests, validation involving DNA are fairly narrow in scope. Where hashing algorithms can take advantage of massive image databases, DNA samples are considerably harder to acquire. Small sample sizes make testing ineffective at catching most errors, especially for specialized populations like Hasidic Jews.[62] This is especially troubling since some errors in STRmix were only found after it was used in thousands of cases.[63]

Limited testing narrows the scope of validation studies in other ways as well. While the likelihood ratios used in TrueAllele are a mathematically sound concept, they rely heavily on assumptions about DNA profiles.[64] These assumptions must be empirically verified to be legitimate, so when TrueAllele is given samples containing animal DNA or more people than was present in validation studies, the usefulness of validation studies is further limited.

While source code disclosure on its own will not allow defendants identify all errors arising from these situations, it can help catch some missteps as developers and prosecutors continue to apply these validation studies beyond their intended scope.

IV. The Risks of Disclosure

A. Financial Harms

In cases with DNA mixture analysis programs and risk prediction algorithms, the primary concern being balanced against the potential insights from source code disclosure is the

---

[62] Stephanie J. Lacambra, Jeanna Matthews, and Kit Walsh. *Opening the Black Box: Defendants' Rights to Confront Forensic Software* https://lin-web.clarkson.edu/~jmatthew/publications/FINALARTICLE_pg28-39_Lacambra_Forensic_Software_May_2018_07102018_608BCX.pdf

[63] *Id.*

[64] Jill Presser and Kate Robertson. *AI Case Study: Probabilistic Genotyping DNA Tools in Canadian Criminal Courts.* Law Commission of Ontario: Toronto, ON, Canada (2021).

developer's proprietary interest in their software. These developers have argued in court that the disclosure of their products would be "financially devastating."[65]

However, these risks may be overblown. Source code leakage is not uncommon in the technology sector. In 2013, the source code for several Abode products was leaked.[66] In 2018, Snapchat's source code and a portion of Apple's Operating System went public.[67]

In addition, recent research challenges the connection between these leaks and severe economic harms. Researchers have found that the leakage of various trade secrets – including source code – do not cause a lasting, statistically significant impact to the victims.[68] Cybercrime literature generally supports these results, finding a surprisingly minimal response to data theft with limited statistical significance.[69] While the risk does increase for smaller firms with fewer trade secrets,[70] a more critical analysis of these developer's risks appears to be necessary.

All this analysis also assumes that a leak happens in the first place. Protective orders, sealing orders, and limiting disclosure to expert witnesses can all limit the risk of leaks in the first place. TrueAllele's competitor STRmix has a policy of limiting disclosing to expert witnesses, which may be what allows it to provide its source code and remain financially viable.

---

[65] *Supra* note 3; *see also* "When a Computer Program Keeps You in Jail" https://www.nytimes.com/2017/06/13/opinion/how-computers-are-harming-criminal-justice.html (TrueAllele has submitted affidavits alleging that disclosing source code to defense attorneys would cause "irreparable harm.")

[66] *Adobe Addressing Massive Data Breach, Source Code Leak* https://www.crn.com/news/security/240162259/adobe-addressing-massive-data-breach-source-code-leak.htm

[67] *Snapchat Code Leaked and Posted to Github* https://www.vice.com/en/article/ywkqew/snapchat-code-leaked-online-github-removed; *Apple iBoot leak was an inside job, and the hacker has more iOS source code* https://www.techrepublic.com/article/apple-iboot-leak-was-an-inside-job-and-the-hacker-has-more-ios-source-code/

[68] Nicola Searle and Andrew Vivian, *Surprisingly Small: The Effect of Trade Secret Breaches on Firm Performance*. (2021).

[69] Alessandro Acquisti, Allan Friedman, and Rahul Telang, 2006. *Is there a cost to privacy breaches? An event study*. ICIS 2006 Proceedings, 94.; *see also* Richardson, V., Smith, R. and Watson, M., 2019. *Much Ado about Nothing: The (Lack of) Economic Impact of Data Privacy Breaches*. Journal of Information Systems, 33(3), pp.227-265.; Gilles Hilary, Benjamin Segal, and May H. Zhang. *Cyber-risk disclosure: Who cares?*. Georgetown McDonough School of Business Research Paper 2852519 (2016).

[70] Anthony Arundel. (2001). *The relative effectiveness of patents and secrecy for appropriation*. Research Policy, 30(4), 611–624. https://doi.org/10.1016/S0048-7333(00)00100-1

STRmix is in a very similar situation to TrueAllele, and as such its willingness to disclose seems to substantially undermine TrueAllele's position.

What would be much more likely to hurt forensic software developers is if judges and defendants begin to find errors in developers' code. Having their products deemed inadmissible in court would have severe financial consequences for the developers. In short, developers can only lose by having their products placed under scrutiny, but it seems the greater risk is from the scrutiny itself, rather than a leak.

B. Harms to Product

While scholarship has focused largely on programs designed to connect suspects to crimes, such as DNA Mixture Analysis Software, some automated techniques like hashing algorithms serve to identify crimes. The developers of these programs have raised another issue in court: that the disclosure of their code could allow criminals to modify their behavior and evade detection.[71] This concern has primarily been raised in the context of hashing algorithms.[72]

TLO, for example, argued that disclosing information about their Child Protection System (CPS) would "significantly degrade the usefulness of CPS as a worldwide investigative tool, as well as compromis[e] numerous ongoing criminal investigations."[73] In doing so they contradict scholarship arguing that security is increased through open-source systems.[74] In

[71] *Supra* note 11, 13, 14
[72] Olivia Solon. *Inside the surveillance software tracking child porn offenders across the globe* https://www.nbcnews.com/tech/internet/inside-surveillance-software-tracking-child-porn-offenders-across-globe-n1234019
[73] *Supra* note 11
[74] Jaap-Henk Hoepman and Bart Jacobs. *Increased security through open source*. Communications of the ACM 50.1 (2007): 79-83.; George Lawton.*Open source security: opportunity or oxymoron?*. Computer 35.3 (2002): 18-21.; Christian Payne. *On the security of open source software*. Information systems journal 12.1 (2002): 61-78.

cryptography there is an applicable concept called *Kerckhoffs' Principle*, the fundamental principle that the security of a system should not depend on keeping the algorithm secret.[75]

There are several benefits to this approach. Systems are generally considered more secure if they require fewer and simpler secrets to be kept. Systems should also be designed to fail gracefully.[76] In addition, most organizations do not have the expert manpower to adequately test private algorithms. Making algorithms public would then help make it less likely that a vulnerability is missed.

While hashing systems like CPS are not designed to be public, this is not due to an immutable aspect of hashing. Algorithms can be public and follow *Kerckhoffs' Principle* by using "private keys." So long as the private key remains secret, these algorithms can be both public and highly secure. Already hashing methods with private keys have been published,[77] meaning that they can be adopted by organizations like TLO.

So, while some current hashing systems might be compromised by disclosure, the loss of effectiveness would likely be temporary and the robustness of hashing algorithms may even be improved.

V. A Path Forward

As the Criminal Justice System becomes more and more automated, it is crucial that the algorithms being used function properly. As expert witnesses become more limited in their expertise, the "circumstantial" issues surrounding source code will become more important.

---

[75] Sasa Mrdovic and Branislava Perunicic. *Kerckhoffs' principle for intrusion detection*. Networks 2008-The 13th International Telecommunications Network Strategy and Planning Symposium. IEEE, 2008.
[76]Charles Mann. *HOMELAND INSECURITY One of the nation's top security experts-Bruce Schneier-warns that the nation's approach to protecting itself is all wrong, and could actually make America more vulnerable than ever*. Atlantic Monthly 290.2 (2002): 81-103.
[77] Rosario Gennaro et al. *Publicly Evaluatable Perceptual Hashing*. International Conference on Applied Cryptography and Network Security. Springer, Cham, 2020.

Under *Daubert*, Courts almost universally accept validation studies as sufficient for admissibility, despite validation studies missing a wealth of critical errors in FST, STRmix, and New Jersey's breath testing devices. While courts have had more mixed decisions regarding discovery, they still tend to accept developers' arguments regarding the merits of disclosure. In response to these arguments, we find that access to source code catches errors that other methods do not, developers' financial risk is likely limited, and the leakage of hashing algorithms is not necessarily damaging. These considerations indicate that current standards are failing to properly gatekeep the scientific evidence from software, or to balance the needs of developers and defendants. Given shortcomings of alternatives such as industry standards and other formal methods, increased requirements for disclosing source code seem necessary to preserve the legitimacy of algorithms in the courtroom.