Privacy When Form Doesn't Follow Function

Roger Allan Ford
University of New Hampshire
roger.ford@law.unh.edu

Privacy When Form Doesn't Follow Function

Scholars and policy makers have long recognized the key role that design plays in protecting privacy, but efforts to explain why design is important and how it affects privacy have been muddled and inconsistent. This article argues that this confusion arises because "design" has many different meanings, with different privacy implications, in a way that hasn't been fully appreciated by scholars.

Design exists along at least three dimensions: process versus result, plan versus creation, and form versus function. While the literature on privacy and design has recognized and grappled (sometimes implicitly) with the first two dimensions, the third has been unappreciated. Yet this is where the most critical privacy problems arise. Design can refer both to how something looks and is experienced by a user—its form—or how it works and what it does under the surface—its function. In the physical world, though, these two conceptions of design are connected, since an object's form is inherently limited by its function. That's why a padlock is hard and chunky and made of metal: without that form, it could not accomplish its function of keeping things secure. So people have come, over the centuries, to associate form and function and to infer function from form.

Software, however, decouples these two conceptions of design, since a computer can show one thing to a user while doing something else entirely. Some of the most pervasive privacy problems, like online tracking and profiling, stem from this misalignment between form and function, since companies can collect, use, and disseminate information without any formal indication that that will occur. Recognizing this third dimension of design, then, can help policy makers identify the most likely privacy problems and fashion reforms directed at inducing realignment.

INTRODUCTION 3
I. DESIGN AND PRIVACY 7
A. Privacy by Design 7
B. Design and Privacy Law Beyond Process 14
II. THE DIMENSIONS OF "DESIGN" 14
A. Process versus Result 14
B. Plan versus Construction 18
C. Form versus Function 21
III. PRIVACY WHEN FORM DOESN'T FOLLOW FUNCTION 24
A. Form versus Function in the Physical World 25
B. Form versus Function in the Digital World 30
IV. IMPLICATIONS AND APPLICATIONS 34
A. Deceptive Design as Form/Function Mismatch 34
B. Unfair Design as Form/Function Mismatch 37
C. Insecure Design as Form/Function Mismatch 40
CONCLUSION 42

INTRODUCTION

Scholars and policy makers have long grappled with the relationship between technology and privacy. Most agree that technologies like the internet, smartphones, and even photography reduce individual privacy, but there is no consensus about how to respond. There is widespread agreement, though, that "design" plays a central role. Since the 1990s, policy makers and regulators (largely outside of the United States) have embraced "privacy by design" as a tool for building privacy protections into new products and services. More recently, the European General Data Protection Regulation, which went into effect in 2018, requires a data controller to implement "data protection by design and by default."[1] And scholars in

1. Regulation (EU) 2016/679 of the European Parliament and the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation), 2016 O.J. L 119/1, art. 25 ("GDPR").

fields as diverse as law and human-computer interaction have unpacked how design can help address the problem of diminished privacy.[2]

Despite this consensus that design matters for privacy, there has been much less agreement about the specifics—why design is important for privacy, how design and privacy interact, and what to do about it. This disagreement has held up progress addressing privacy problems and has allowed interested industry players to coopt the "privacy by design" label for their own ends.

This article argues that this disagreement stems from a failure to grapple with the many meanings of "design" and the ways that these meanings affect privacy. Design, like privacy itself, can mean many different things in different contexts, and privacy scholars writing about design have used different definitions, sometimes in the same piece and often without realizing it, muddling their analysis.

These definitions of design exist on three different dimensions. First, design can be a process or a result of that process. An architect designs a house—design as process—and embodies that design in blueprints—design as result. In the former, design is a verb; in the latter, a noun. Second, design can have different relationships to the object being designed. Design can be a plan or a planning process—design as plan—existing before separate from the object, as when an architect designs a house. Or it can be the object itself and the process of constructing that object—design as construction—as when an artist creates a painting. Or the two can be entwined, as when a website evolves through an iterative process. And third, design can represent different perspectives or parts of an object. It can refer to its look or visual representation of an object—design as form—or to the way it works or how users interact with it—design as function. All these things are "design" in different contexts to different speakers.

Not all of these different conceptions of design have the same effect on privacy. Advocates of privacy by design have focused on design as process and design as plan, arguing that addressing privacy earlier in a product or service's planning

2. See, e.g., Woodrow Hartzog, Privacy's Blueprint: The Battle to Control the Design of New Technologies (2018); James Pierce et al., An Interface Without A User: An Exploratory Design Study of Online Privacy Policies and Digital Legalese, 2018 Proc. Designing Interactive Sys. Conf. 1345; Richmond Y. Wong et al., Eliciting Values Reflections by Engaging Privacy Futures Using Design Workbooks, 1 PACM Human-Comp. Interaction art. no. 111 (2017); Ari Ezra Waldman, Designing Without Privacy, 55 Hou. L. Rev. 659 (2018); (others).

cycle will lead to fewer privacy problems. Others have focused on the results of those processes, arguing that the way something looks and works determines whether it causes privacy problems. And scholars and regulators have struggled with how address new kinds of products and services that continually evolve, where there is no clear division between planning and execution. Focusing on design as creation could be a better match for these products, especially on the internet.

Even breaking it down in this way, though, it's unclear why some products and services cause large privacy problems while others don't, or what role design plays in that outcome. This article argues that the key lies in the third conception of design, form versus function. Modern privacy problems usually arise in the context of new technologies: the growth of the internet and web tracking, the use of smartphones with location tracking, even instant photography in the late 1800s. What makes new technologies important is that they enable new and different functionality—critically, often without new forms. This is especially true of software, which decouples form and function nearly entirely. When this happens, the single best way that people have to avoid privacy losses—a visceral warning of an imminent loss—never comes.

Consider a padlock. In the physical world, the form and function of a padlock are linked. A padlock is thick, heavy, and made of hardened metal because these things are necessary for it to work, so it can keep things secure without being cut off or ripped off. A padlock with different form—say, a thin and elegant padlock, or one made of paper or wood or fabric or aluminum foil instead of thick metal—wouldn't accomplish that function; it wouldn't keep things secure. And so people have, over the course of decades or centuries, come to associate certain forms with their associated functions. But in the world of software, these things are unlinked; an app can display a padlock to indicate that something is secure whether or not it's actually secure. So when web browsers started using a padlock icon in the address bar to indicate a secure connection, some fraudulent sites started using fake padlocks as their website icons to pretend that they were secure.[3] And when hardware and software come together, the potential for a mismatch between form and function is heightened further, as when the maker of a fingerprint-operated "smart

3. E.g., Phishing Scam Website solution-place.ru, Online Threat Alerts, https://www.onlinethreatalerts.com/article/2013/2/9/phishing-scam-website-solution-place-ru/ (Feb. 12, 2013).

padlock" was surprised to learn that it could be opened with a screwdriver.[4]

Many of the most pressing privacy issues stem from this sort of form/function mismatch. Pervasive online tracking is a good example. Websites can track people due to the functional design of web browsers. Browsers store cookies, which are snippets of text sent by a server, and websites use this to assign users unique ID numbers. And data brokers and ad networks can track people across websites because browsers are fine with the code for a webpage embedding elements from third parties. So if both the New York Times and Washington Post websites use the same provider to serve ads, then that provider can place its own cookie on a user's computer and see that that user reads both the Times and the Post. Ad networks that work with a lot of websites, then, can build detailed profiles of users and monetize those profiles with tailored ads. But none of this is remotely evident from the form of the web browser; it all takes place below the surface, hidden to the user. It's as if the padlock, in addition to securing one's possessions, had a hidden camera tracking everything its user does. Instead of the object's form flowing from its function, the form conceals a hidden function.

Looking to this third dimension of design, then, can help policy makers identify the products and services most likely to lead to privacy problems. But it also provides an obvious mechanism for fixing those problems: If privacy problems arise when form and function misalign, then reforms should focus on inducing realignment. There are lots of ways to work toward this goal—incentives, mandates, enforcement under the FTC Act—but lack of clarity about the goal has hindered progress.

This article has four parts. Part I reviews the role of design in privacy law and privacy theory, concluding that design-centered accounts of privacy have largely failed. Part II argues that this failure is rooted in the incompleteness of these accounts' definition of design. It unpacks three dimensions of design—process versus result, plan versus creation, and form versus function—and the roles that each kind of design play in determining the level of privacy that people enjoy. Part III focused on the third of these dimensions, arguing that many of the most substantial privacy

4. Mark Frauenfelder, High Tech Lock is 'Invincible to People Who do Not Have a Screwdriver,' Boing Boing, https://boingboing.net/2018/06/15/high-tech-lock-is-invincible.html (June 15, 2018).

problems arise when form and function are misaligned. Software, in particular, nearly wholly decouples form and function, making it easy for developers to invade privacy in ways that have no parallel offline. Part IV applies the mismatch theory to three recurring kinds of privacy problems—deceptive design, unfair design, and insecure design—and discusses its implications for solving these problems.

I. DESIGN AND PRIVACY

Scholars and policy makers have long recognized the importance of design to privacy, though what they mean by design has varied, as have its effects and implications for privacy law and policy.

A. Privacy by Design

Invocations of design as being important to privacy date at least to the 1990s, when regulators were increasingly focused on finding ways to counter privacy problems arising from new technologies like the internet.[5]

In 1995, for instance, the Dutch Registratiekamer (data protection authority) and the Information and Privacy Commissioner of Ontario, Canada conducted a survey and published a joint report on "privacy-enhancing technologies" as a mechanism for countering this trend.[6] The joint report focused on privacy in the consumer context. The regulators started with the observations that many transactions required consumers to identify themselves—so that "an identifiable record of each transaction is created and recorded in a computer database somewhere"—and that that this was so pervasive that it was just accepted as a given by many.[7] But, they

5. E.g., Peter Hustinx, Privacy by Design: Delivering the Promises, 3 Identity Info. Soc'y 253 (2010) (recounting the origins of "privacy by design" as a concept in the 1990s, by the European Data Protection Supervisor); Ann Cavoukian, Privacy by Design (2009), available at http://www.ontla.on.ca/library/repository/mon/23002/289982.pdf (similar, by the Information and Privacy Commissioner of Ontario, Canada). See also Directive (EC) 95/46 of the European Parliament and of the Council of 24 October 1995 on the Protection of Individuals with Regard to the Processing of Personal Data and on the Free Movement of Such Data, 1995 O.J. L 281/31 (the European Union Data Protection Directive).

6. See Information and Privacy Commissioner, Ontario, Canada & Registratiekamer, The Netherlands, Privacy-Enhancing Technologies: The Path to Anonymity (1995), available at http://www.ontla.on.ca/library/repository/mon/10000/184530.pdf (hereafter "Joint Report").

7. Id. at vol. 1, sec. 1.0.

observed, this made it basically impossible to conduct transactions that were both anonymous and secure. And since databases were increasingly easy to connect to each other, the privacy losses from this data recording would increase.

To solve this problem, the report called for a "a paradigm shift away from a 'more is better' mindset to a minimalist one" that would minimize collection while still fulfilling data collectors' needs.[8] And it looked to technology as the means for effecting that shift. The report called for the adoption of "identity protectors"—basically, software under the control of users that lets them provide pseudonymous information to systems instead of real, personally identifiable information.[9] So, for instance, a physician could store patient information under patients' pseudo-identities instead of their real identities. This would, the report posits, let third parties like researchers have access to patient data without compromising patient privacy.[10] (There are obvious issues with this on both ends of the privacy/utility spectrum. Health information can be highly identifying even without labels. And using different pseudo-identities for different providers, as the report proposes, would prevent researchers from drawing potentially valuable correlations; this is one reason that so much effort is being invested in electronic health records and health applications of big data. But this was the mid-1990s, before these issues had been well developed.)

The report was surprisingly blasé about *why* technology was leading to reductions in privacy or how technology developers' needs and incentives might affect privacy. The "key question," the report asserts, is how much personal information "is truly required for the proper functioning of the information system" used to conduct a transaction.[11] When one asks that question from the outset, then privacy will inevitably be respected: "If the client, the systems designer and supplier ask this question right from the start, privacy is sure to be addressed."[12] The only remaining issue is how to implement privacy protection with an identity protector, and for that, the report explains that it "may be done in a number of ways, usually

8. Id. at vol. 1, sec. 1.3.

9. Id. at vol. 1, sec. 1.6.

10. Id. at vol. 2, app'x B.

11. Joint Report, supra note 6, at vol. 1, sec. 1.3.

12. Id. at vol. 1, sec. 1.7.5.

involving advanced encryption techniques (best left to systems designers and technical staff)."[13] Whether a system *needs* access to personal information is a function of the system's goals—a system used for targeted advertising needs more access to personal information than one used for untargeted advertising—but the report offers no guidance on identifying or developing these goals. Instead it only briefly notes that one challenge would be "the reluctance of both public and private sector organizations that wish to collect *more,* not less, identifiable information," though it minimizes the challenge by positing that organizations will come to understand "the benefits of collecting less."[14]

Instead of focusing on these sorts of system goals, the report's key takeaway concerned process and planning. It explained that creating an identity protector "must [ ] be a crucial part of the design phase" of a technological system[15] and surveyed technology providers about available privacy technologies that could be incorporated in this way.[16] And its first two recommendations focused on ways to operationalize privacy-conscious design processes, with one suggesting the development of "international information systems design standards" and the other suggesting that in the design of a new system, the collection and retention of personal information should be kept "to an absolute minimum."[17]

This focus on process and planning carried over as "privacy by design" evolved into a label for a proactive approach to privacy in businesses and engineering teams. Ann Cavoukian, who had been Deputy Commissioner when the joint report was written and who claims credit for originating privacy by design when she later served as the Information and Privacy Commissioner of Ontario, explains that the approach is "proactive rather than reactive" and "anticipates and prevents privacy intensive events before they happen."[18] It does this by providing ways of "influencing, shaping, and regulating" the architecture of a technology system to reflect

13. Id.

14. Joint Report, supra note 6, at vol. 1, sec. 3.0.

15. Id. at vol. 1, sec. 1.7.5.

16. Id. at vol. 1, sec. 2.

17. Id. at vol. 1, sec. 3.1.

18. Ann Cavoukian, Privacy by Design, Info. & Privacy Comm. of Ontario (revised Sept. 2013) ("Privacy by Design Overview"), available at https://www.ipc.on.ca/wp-content/uploads/2013/09/pbd-primer.pdf.

important values and goals, including privacy.[19] These ways, Cavoukian explains, should take the form of organization-wide privacy structures, starting with executive-level commitments, since piecemeal approaches are more likely to lead to spotty and inconsistent privacy protections.[20] Cavoukian offers examples of tools that can be used to implement an organization-wide privacy commitment, like impact assessments, risk-management processes, and external audits and certification.[21] Whatever tools an organization uses, the theory goes, having a privacy commitment and enforcement mechanism in place in advance should lead to better privacy outcomes, since they reduce the need for teams to reinvent the privacy wheel for each new product or feature and make it difficult or impossible to miss the privacy implications of a system's architecture.

In recent years, scholars and policy makers have taken up Cavoukian's call. Ira Rubinstein, for instance, proposed ways that regulators could encourage companies to adopt privacy by design, which he defined as "the idea that in designing information and communications technologies, building in privacy from the outset achieves better results than bolting it on at the end."[22] In doing so, Rubinstein analyzed the reasons that companies might underinvest in developing such comprehensive privacy programs without government intervention. He posited, for example, that because many of the benefits of stronger privacy protections are uncertain or occur later in time—like not having to pay the costs of a data breach—firms may "lapse into a reactive mode," delaying investment until forced to do so.[23] But if, for example, the FTC took on a strategic enforcement strategy and sought out cases that would make useful precedent, then firms might recognize the need to incorporate privacy throughout their corporate structures and product-development processes.[24]

---

19. Ann Cavoukian, Privacy by Design in Law, Policy and Practice: A White Paper for Regulators, Decision-Makers and Policy-Makers 13, Info. & Privacy Comm. of Ontario (Aug. 2011), available at http://www.ontla.on.ca/library/repository/mon/25008/312239.pdf

20. Id. at 14.

21. Id. at 15–16.

22. Ira S. Rubinstein, Regulating Privacy by Design, 26 Berkeley Tech. L.J. 1409, 1410 (2011).

23. Id. at 1432. Rubinstein also points to other reasons to expect underinvestment in privacy, such as information asymmetries and other classic market failures. Id. at 1431–32.

24. See id. at 1446–47; see also Daniel J. Solove & Woodrow Hartzog, The FTC and the New

In recent years, the European Union has taken up the mantle of privacy by design. It has incorporated a version of the concept into Article 25 of the General Data Protection Regulation, which requires data controllers to provide for "data protection by design and by default."[25] And six days after the GDPR went into effect, the EU Data Protection Supervisor issued a "preliminary opinion on privacy by design," containing recommendations to EU member states on ways to implement and enforce the Article 25 obligations.[26] The opinion rooted many privacy problems in a gap between two distinct business cultures: the "legal compliance discipline managed by lawyers" and the "dynamic innovation process driven by business managers and engineers … who are ultimately responsible for the design and implementation of the processes and systems that govern the real functioning of the organisation."[27] When faced with the difficult task of complying with shifting and inherently vague privacy rules, the result too often was that the lawyers lost the internal fight.

To counter this dynamic, Article 25 requires data processors to take affirmative steps to protect data both when they are deciding how data will be processed (a term of art in European privacy law that encompasses basically anything one might do with data, including collecting, storing, altering, using, destroying, or disseminating it[28]) and when that processing occurs. These steps must be ones that, "by default," minimize data processing so that only the data that are necessary for a specific purpose are processed. And the article calls out specific techniques for doing so, like pseudonymisation—the same technique highlighted by the Joint Report 23 years earlier.[29]

Article 25 follows Article 24's basic rule setting forth data controllers' core obligation, that they "implement appropriate technical and organisational measures to ensure and to be able to demonstrate that processing is performed in

---

Common Law of Privacy, 114 Colum. L. Rev. 583 (2014) (arguing that FTC enforcement actions and settlements amount to an effective common law of information privacy).

25. GDPR, supra note 1, at art. 25.

26. European Data Protection Supervisor, Preliminary Opinion on Privacy by Design, Opinion 5/2018 (May 31, 2018) ("Preliminary Opinion").

27. Id. at ¶ 14.

28. See GDPR, supra note 1, at art. 4(2).

29. See Joint Report, supra note 6.

accordance with this Regulation," i.e., that they follow the law.[30] This regulatory structure is revealing. Article 24 states the goal—that controllers implement measures to protect privacy—while Article 25 explains how—by using privacy-protecting tools like pseudonymization and by doing so during both planning and operational phases of a project. This illustrates a central assumption underlying privacy by design: that although the end goal is protecting people's privacy, the most effective regulatory targets are the planning stages and processes leading up to that goal.

In the United States, the Obama-era Federal Trade Commission likewise urged companies to adhere to privacy-by-design principles, albeit as just one of three core privacy principles and without the force of law.[31] The FTC also took a development-cycle view of privacy by design, summarizing the "baseline principle" as "Companies should promote consumer privacy throughout their organizations and at every stage of the development of their products and services."[32] To do so, the FTC asserted, requires adopting a set of substantive privacy protections like data security, collection limits, and reasonable retention practices and accuracy rules,[33] and implementing those protections procedurally through "comprehensive data management procedures throughout the life cycle of their products and services."[34]

A notable consequence of these visions of privacy by design is that they make a choice among competing models of product development. They assume that products are developed through a multi-stage process of planning followed by execution or creation. Indeed, the Preliminary Opinion took this assumption about how businesses are run and turned it into a normative ideal, asserting the "fact"

30. See GDPR, supra note 1, at art. 24.

31. See FTC, Protecting Consumer Privacy in an Era of Rapid Change: Recommendations for Businesses and Policymakers 22–34 (Mar. 2012), available at https://www.ftc.gov/sites/default/files/documents/reports/federal-trade-commission-report-protecting-consumer-privacy-era-rapid-change-recommendations/120326privacyreport.pdf; Edith Ramirez, Privacy By Design and the New Privacy Framework of the U.S. Federal Trade Commission, Privacy by Design Conference, Hong Kong (June 13, 2012), available at https://www.ftc.gov/sites/default/files/documents/public_statements/privacy-design-and-new-privacy-framework-u.s.federal-trade-commission/120613privacydesign.pdf.

32. FTC, supra note 31, at 22.

33. Id. at 23–30.

34. Id. at 30.

that data processing "should always be the outcome of a design project."[35] But there are other ways to do it. Products can evolve as customers' needs evolve or as a seller's understanding of those needs evolve; products can be jointly created by buyers and sellers; products can be created by large, uncoordinated groups working without any central planning or direction; products can be created by happy accident. So there are ways that things can be created while not being the outcome of any sort of design project.

Consider two encyclopedias: Britannica, which is written by a small group of full-time editors and a larger group of contributors, and Wikipedia, which is written by many more volunteer editors, working around the world, largely without central direction or coordination. Britannica is the result of the sort of intentional design project that the Preliminary Opinion discusses, with editors deciding what topics need coverage, how much space to devote to a topic, and so forth, based on the encyclopedia's goals, market, and so forth. Wikipedia, not so much. Wikipedia is designed in the sense that someone created the wiki software and the Wikipedia logo and so forth. But Wikipedia's *content* and overall structure are not designed in this way; no central authority decides that the encyclopedia is heavy on sci-fi content but light on opera content and directs contributors to expand the pages for operas. Instead of reflecting a design process, Wikipedia's content reflects the aggregate of its editors' interests, opinions, and beliefs. And this is a legitimate product-development strategy, albeit one that the Preliminary Opinion seems to dismiss as a possibility.

If privacy by design suggests, then, that these sorts of undirected creative processes are bad, at least for privacy, then it is probably right. Undirected creative processes like collaborative crowd sourcing or product evolution can optimize for different things, like the aggregate views of the crowd or the needs of a product's consumers. But privacy isn't usually one of the critical factors guiding product development and influencing its chances of success. Privacy by design can correct this by injecting privacy as a consideration from the beginning, but how to do so is unclear, which is why it has been criticized for its vagueness and for failing to offer specific tools or technologies or recommendations for incorporating privacy throughout the design process.

35. Preliminary Opinion, supra note 26, at ¶ 27.

B. Design and Privacy Law Beyond Process

[And now, the piece of the draft that I haven't had time to write yet. It'll review literature on how the designs of products, especially software, lead to privacy problems.]

Efforts to characterize the relationship between privacy and design have sometimes gone beyond the privacy-by-design framework. Probably the most prominent example is Woody Hartzog's book *Privacy's Blueprint: The Battle to Control the Design of New Technologies,*[36] which offers a framework for governing privacy by directly regulating product designs rather than putting the onus on consumers and technology users to protect themselves. * * *

Likewise, scholars have focused on product designs and their privacy implications. Ari Waldman and James Grimmelmann, for instance, have studied how Facebook uses design tools to encourage users to share information.[37] * * *

II. THE DIMENSIONS OF "DESIGN"

Design is a fraught concept because it has so many meanings. Scholars have struggled to explain the relationship between privacy and design in part due to this ambiguity; privacy is a concept that is notoriously hard to define, but design may be just as difficult. Design has different meanings, in different contexts, with different implications for people and privacy. Only by unpacking these different meanings, then, can design be harnessed to protect privacy.

This Part considers three dimensions along which conceptions of design vary—process versus result, plan versus creation, and form versus function—and the implications of these conceptions for privacy.

A. Process versus Result

Sometimes, design is a process, while other times, design is the result of that process. Sometimes design is a verb; sometimes it is a noun.

---

36. Woodrow Hartzog, Privacy's Blueprint: The Battle to Control the Design of New Technologies (2018).

37. Ari Ezra Waldman, Privacy, Sharing, and Trust: The Facebook Study, 67 Case Western Res. L. Rev. 193 (2016); Ari Ezra Waldman, Manipulating Trust on Facebook, 29 Loyola Consumer L. Rev. 175 (2016); James Grimmelmann, Saving Facebook, 94 Iowa L. Rev. 1137 (2009).

Consider the design of a car. In some cases, design refers to the process of planning and creating the car: "designing" the car. A design team works for years to create a new car model, brainstorming, iterating, and refining along the way. In this conception, design is a process of problem solving in an environment of constraints.[38] Some of these constraints are obvious: market research about what consumers are looking for, how the car fits into the manufacturer's lineup or compares to the competition, the expected price point and profit margins, and so forth. Others are less obvious but still important. A designer might want to make a beautiful car, or one that looks tough or is reminiscent of a historical model. They might want to bring back the rotary engine or see if they can create a flatter torque curve or a quieter and more luxurious interior. They might want to create a car that handles well in the snow or one that hugs the road in sharp turns. Some of these constraints can be concrete, well-defined, and inflexible, like safety and environmental regulations. Others might be more aspirational or vaguely defined, like a company's values or its heritage. Usually, no one factor dominates; different constraints will inevitably be in tension with each other and will evolve and be traded off over the course of the design process. In this conception, design is an extended process that eventually comes to an end, producing a result: a new car.

Other times, design refers to the result itself and its features and characteristics. When car reviewers talk about the design of a new car, they don't mean the design process; they mean the car, and how it looks and feels and drives. But the design result still reflects a solution to the problem of satisfying a set of constraints by balancing different goals. The design of a sports car might include the fact that it is bright red, that it is low to the road, that it has a powerful engine, that it has a carbon-fiber roof, that it has a swoopy, aerodynamic shape, that it costs $100,000. The design of a family minivan might instead include the fact that it has an automatic sliding door, that it has three rows of seats, that it has a roof rack, that it has a built-in video-game system in the back, that it costs $25,000. The designs are different because the design processes solved for different constraints. One set of designers produced a car for a wealthy buyer who wants to drive fast and show off, but also maybe a car that helped develop new engine technologies that will work their way down to the maker's other cars in the future. The other set of designers

38. Cf. Mike Monteiro, Design is a Job (2012) ([quote about design being problem solving]).

produced a car for families who want to take the kids to soccer practice in comfort.

This may seem like a difference of semantics, that since the design process leads to the design result, there is no meaningful difference between the two. But they have different implications. Different design processes work in different ways, leading to different kinds of outcomes. And different resulting design results describe fundamentally different products, with different characteristics and behaviors and effects on their users and others.

The effects of different design processes can be profound; this is the root of the field of design methods.[39] There is no one way to design a thing, so different designers use different methods, which give different results. One method might work in stages, working to solve a design problem. The design theorists Kees Dorst and Judith Dijkhuis describe this approach as a process of "rational problem solving."[40] This process includes steps like listing known constraints, analyzing them to determine potential solutions, testing, iterating, and logically searching for a solution to the problem. It is a structured, rigorous, planned process using logic, reason, and evidence to come to a result. In contrast, another approach might be far more improvised. It might blending different steps of the design process, moving quickly between generating new ideas and implementing and evaluating those ideas, and relying on creativity, brainstorming, emotion, and inspiration. Dorst and Dijkhuis call this "reflection in action."[41] Compared to the rational model, it can be quicker and find more innovative solutions to problems, but can also lead to more problems if important criteria were missed or the creative process went astray.

While the design process matters, since different processes will lead to different results, those effects are indirect. But if the goal is to evaluate the consequences of a design for users, it is the design result that matters most; those effects are felt directly. A better process might give a better result, but if designers can get the better result without the better process, then the process doesn't so much matter.

Privacy is a good illustration of this. Different products and services produce different privacy outcomes because of their designs, in the results sense rather than

39. E.g., John Chris Jones, Design Methods (2d ed. 1992).

40. Kees Dorst & Judith Dijkhuis, Comparing Paradigms for Describing Design Activity, 16 Design Studies 261, 266 (1995).

41. Id. at 271.

the process sense. When Snapchat promised users that their photos and videos would be protected and could not be saved by their recipients or viewed for longer than the amount of time set by the sender, the promise failed because the design of the app provided for loopholes.[42] Photos were not encrypted; they were stored on the device outside of the app sandbox; the service's API performed no authentication and so would allow any third-party app to connect. These features and characteristics of the app—of its design—led to privacy losses. Had Snapchat used a different design process, those features and characteristics might not have been the same, but it is the design result that caused problems, not the design process.

Why, then, the privacy literature's heavy emphasis on process? I think there are two main reasons. One is that regulating process is often much easier than regulating result. There are many available process reforms, like telling companies that they have to evaluate privacy throughout the design process, or that they have to have some specific privacy-protection infrastructure like a chief privacy officer, or that they have to engage in periodic privacy assessments. These reforms can be done without grappling with the difficult questions of exactly what privacy interests must be protected or how strongly. And they can be context-neutral: the same process reforms might be equally applicable and relevant for a software company, a retailer, or an insurance company. So there may be less need to delve into the specifics of the business and figure out exactly what about the design result of its product led to privacy losses or could do so.

The other reason the privacy literature emphasizes process is that many privacy failures—though certainly not all—likely do stem from failures of process. In many cases, for instance, privacy losses stem not from intentional design decisions, but from mistakes: buggy code, out-of-date security software, inadvertently leaving a laptop in a car. Process can reduce the chances of these mistakes, just as following a checklist can reduce the chances of mistakes by surgeons and airline pilots.[43] In other cases, privacy losses arise because privacy issues were considered too late in the design process, once other decisions had been made and baked into a project. This could happen because privacy is not considered by the design team to be an important issue, for instance, or because different phases of the design process are

42. In re Snapchat, Inc., FTC Complaint (May 8, 2014).

43. Atul Gawande, The Checklist Manifesto: How to Get Things Right (2009).

done by different groups. A project started by a team of engineers, like at a new startup, might not have lawyers or other privacy professionals involved until far later. In cases like these, simply considering privacy as one of the design constraints throughout the design process—and finding ways to make companies do so—could have a meaningful effect.

Still, if the ultimate question is the privacy implications of different products and services, then it is the characteristics of those products and services that matters more than their design processes. If regulators had good mechanisms for regulating design results directly, or their effects on privacy, then there would be little reason not to prefer that approach.[44] But that's hard to do, especially in technology industries, where best practices evolve on an hourly basis.[45] In those cases, regulating process may be the next best thing.

B. Plan versus Construction

Design can also mean different things temporally. Sometimes, design refers to a planning process that occurs before construction and use of an object or system, or a plan that is created before those things can happen. Other times, design and construction occur at the same time and cannot readily be distinguished. And other times, a product evolves over time, with design steps and construction steps overlapping and intermingling.

Unlike the process/result dimension of design, which provides two different ways of thinking about the design of the same sorts of things, these temporal conceptions of design more often refer to different categories of design tasks. Some types of design lend themselves to separate planning and construction stages or can only work through separate stages. Consider the design of a building. A builder cannot reasonably begin construction while an architect is still designing the building. If the builder has already poured a foundation and assembled the building's frame when the architect decides to move an exterior wall out by six feet, then the

44. Indeed, this is the premise of strict liability in products-liability cases: by forcing manufacturers to internalize the cost of injuries inflicted by their products, strict liability encourages them to take every cost-effective precaution without having to dig into the details of the product-design process. (Citations about products liability and privacy.)

45. E.g., LabMD, Inc. v. FTC, 894 F.3d 1221 (11th Cir. 2018) (vacating FTC order that LabMD implement a commercially reasonable data-security program as too vague to be enforceable).

process will run vastly over budget or the result will have fatal flaws. So architects develop complete plans for a building before major construction work begins.[46]

Other types of design lend themselves to the collapse of design and construction into one process. Many crafts can work this way: a potter shaping a ceramic pot on a potter's wheel or a glassblower free-blowing a vase might design the pot or the vase as she constructs it rather than having a final design in mind from the beginning. But this also applies to modern design tasks like graphic design and software design, where the product, be it a page layout or an app home screen, may be designed at the same time it is built.

Some of the most modern design tasks blur this line between design and construction. This is especially true on the internet, where products can be continually updated, instead of being locked in before shipping. Consider the design of the Facebook website or the Google search algorithm: there is neither a clear progression from the design phase to the construction phase nor a collapse of those into one process. Instead, the product evolves over many years, with constant changes, both visible and invisible to users. Some of those changes might go through a discrete design phase, for instance by mocking up a new interface and gathering reactions before any changes are implemented. Indeed, Facebook created and released a tool that makes it easier to create those mockups.[47] Others will be built and deployed without any real design phase. And for some changes, the design process effectively happens after construction, when different options are A/B tested against each other, as when Google famously tested dozens of shades of blue to see which one led to the most clicks.[48]

Why does this matter for privacy? If regulators focus on the design process when addressing privacy problems, then their implicit and explicit assumptions

46. There are always exceptions, of course. For major projects, an architect might be present on site in case issues come up that require revisions or clarifications.

47. See Facebook, Inc., Origami Studio, https://origami.design/ (last visited Oct. 7, 2018); Brandon Walkin, Introducing Origami Live and Origami 2.0, Medium, https://medium.com/facebook-design/introducing-origami-live-and-origami-2-0-a68116294e65 (Feb. 24, 2015).

48. Alex Hern, Why Google Has 200m Reasons to Put Engineers Over Designers, The Guardian, https://www.theguardian.com/technology/2014/feb/05/why-google-engineers-designers (Feb. 5, 2014) (reporting that when Google tested different shades of blue for ad links in Gmail, the winning shade led to $200 million a year in added revenue).

about how that process works will affect the regulations that they adopt. And since imposing process controls is quickly becoming the most popular approach for privacy regulation, getting it right is key.

But some process controls will work better than others, depending on a company's design process. For instance, requiring companies to use threat modeling to assess privacy risks when developing new products might seem like a commonsense reform with few downsides. But it assumes a development process that puts design before construction, so it works better for a slow and deliberate development process, like might be used on an important new product or a major change to an existing product. It's less suitable for the more nimble, iterative design process that might be used to update a complex web app, since it would interfere with the speed and flexibility needed for that process to work. In contrast, other process controls like mandatory firmwide training or after-the-fact testing might apply equally well to different sorts of design models.

One obvious way to use process controls to solve privacy problems while still allowing firms to use different design models is to be flexible with the specific controls employed—to let firms choose processes that fit their firm structure and practices while still protecting privacy. The FTC has taken this approach, requiring in several consent decrees and orders that a company adopt a "comprehensive privacy program" that is "appropriate to [the company's] size and complexity, the nature and scope of [the company's] activities, and the sensitivity of the covered information."[49] This reliance on standards rather than rules might be a good approach, since it avoids problems adapting rules for the wide array of corporate structures, sizes, products, design and development processes, user bases, and other characteristics of the companies with access to personal information. But it also provides poor notice of exactly what companies have to do to stay on the right side of the law—a flaw that the Eleventh Circuit found fatal when it vacated an order requiring the medical testing firm LabMD to adopt reasonable data-security practices.[50] And it gives firms enough wiggle room to avoid the specific privacy protections

49. This example is from the Snapchat settlement. See In re Snapchat, Inc., FTC docket no. C-4501, decision and order at 3 (Dec. 23, 2014), available at https://www.ftc.gov/system/files/documents/cases/141231snapchatdo.pdf.

50. LabMD, supra note 45.

that might, in a particular context, greatly benefit consumers but harm firms.

Another solution goes the other direction: Instead of giving firms flexibility, adopt process reforms that apply throughout the lifecycle of a product, from design to construction to operation to termination, or that work equally well for different design approaches. Article 25 of the GDPR, for instance, requires data process to implement appropriate safeguards "both at the time of the determination of the means for processing and at the time of the processing itself"—i.e., both in the design phase and after.[51] Article 25 is in some ways flexible, instructing data processors to "[take] into account the state of the art, the cost of implementation and the nature, scope, context and purposes of processing as well as the risks of varying likelihood and severity for rights and freedoms of natural persons posed by the processing" in determining the appropriate protections.[52] But it is wholly inflexible about when such protections must be taken; Article 25's answer is that they must be implemented throughout "the whole project lifecycle."[53] Such inflexibilities can still limit companies' choice when selecting design models—a company could not, for instance, intentionally choose a process that ignores privacy until the very end of the proves—but they leave companies substantial flexibility to decide *how* to consider privacy at every stage of the product life cycle.

C. Form versus Function

Finally, design can also refer to different parts of a product. Sometimes, design refers to how a product looks—its visual design or form. Other times, design refers to what a product does or how it works—its functional design or function. This is a fundamental distinction between different kinds of design, but it is one

51. GDPR, supra note 1, at art. 25 § 1.

52. Id.

53. Preliminary Opinion, supra note 26, at ¶ 27. The Preliminary Opinion even goes so far as to clarify that although Article 25 refers to "the determination of the means for processing and … the processing itself," and not other stages, this should be read to include the whole lifecycle of a project:

> In general, in project management literature, the "implementation/construction" of the project/system, following the design and preceding its operation, and the "dismissal/transition" of a project/system following its operation are also noticeable project phases with their own specific requirements. Nevertheless there are no reasons to believe that the legislator did not want to refer to the whole lifecycle of a project by just mentioning the design and operational phases.

Id. at n.39.

that the privacy literature has mostly ignored.

Sometimes, design refers solely or mostly to how something looks. This is most obvious when it comes to visual arts and related fields: graphic design, for instance, is purely visual, and fields like marketing and package design have substantial visual elements. But it applies to other kinds of design as well. Apple got a huge amount of attention for redesigning iOS between versions 6 and 7, with the new design eliminating skeuomorphic gradients, shadows, and shading in favor of a flatter, starker aesthetic. The redesign was almost completely visual; the software still worked mostly the same, and users' mental models of how to use the system remained valid. Users still pressed the home button at the bottom of the phone to go to the springboard, a grid of icons for individual apps. They still typed on the same sort of on-screen keyboard and still selected and copied and pasted text in the same way. Lists still scrolled the same way, using the same sort of faux-real-world physics to determine the scrolling velocity and "bouncing" at the end of the list in the same way. But the software looked different, so the change got far more attention than other changes to the functional design of the operating system.

Other times, design refers solely or mostly to what something does and how something works—its function, internal structure, and so forth. Designing processes and procedures is the purest form of this, since there's no visual nature of the product in the first place. The entire field of chemical engineering, for instance, is one of design, focusing on developing methods for producing chemicals and performing chemical reactions at scale. An engineer will work to design a process that produces the desired output while balancing speed, cost, safety, purity, and other design constraints. But aside from the bubbling beakers full of colorful chemicals seen on tv, there's no visual component to the design. And the same is true of designing better algorithms, data storage models, computer processors, car engines, and so forth. And even when an artifact has a visual nature, design can still refer to function. The functional design of an aluminum soda or beer can, for instance, includes elements like a concave domed base, cylindrical sides, crimped top rim, and tabbed opening. These elements are experienced by consumers, but they are designed for functional purposes like increasing the strength-to-weight ratio and making the can mechanically fillable.

There are intermediate types as well. An architect, for instance, might borrow from many traditional styles, like the gothic, renaissance, classical, art deco, and

modern styles. These styles have both visual and functional meanings. Brutalism is a good example: the brutalist style is characterized by the use of concrete surfaces and blocky shapes that create an imposing, bureaucratic, and fortress-like image; it's also often characterized by outer forms that reveal the organization and functionality of the inside space. Or, Giyōfū is a Japanese architectural style characterized by buildings that look like they are built in Western styles but are built using traditional Japanese methods. So an architect's choice of styles and the incorporation of elements from those styles into a design will necessarily constrain and inform both the visual design of the building and its functional design.

Both of these conceptions of design can affect people's privacy. With functional design, this is obvious. Objects are designed to perform functions, and they perform those functions using different means. Some of those functions and means protect privacy; some don't. The basic function of a security camera is to capture information about people; the basic function of a curtain is to shield people from being observed by others. Both objects affect people's privacy due to their fundamental functionality. Likewise, when two objects have the same ultimate function, decisions about how they are structured and how they achieve that function can affect privacy. Messaging software that uses end-to-end encryption will accomplish the goal of enabling person-to-person text communications without harming its users' privacy; messaging software that uses no encryption will cause such a harm. Or, two refrigerators might equally well accomplish the goal of keeping food cold, but the one that has an oleophobic coating on the door handle will better protect privacy than one that lacks such a coating and so accumulates fingerprints.

Privacy law frequently considers design elements that fall on one side or the other of this line between form and function, though it does not routinely consider the distinction itself. Data-security cases, for instance, are often about functional designs that fail to protect privacy, as when data is stored on computers with buggy and out-of-date security software. Similarly, the FTC has repeatedly recognized ways that the visual design of a product can deceive consumers, as when that design uses a padlock to suggest that a connection is secure, when it is not. But the distinction itself is worth considering more deeply, because it is frequently at the root of some of the thorniest privacy problems, as discussed in the next Part.

III. PRIVACY WHEN FORM DOESN'T FOLLOW FUNCTION

The three dimensions of design discussed in Part II are all relevant to the question of privacy regulation, but the third dimension plays a more fundamental role than the other two. Form and function don't merely exacerbate or alleviate privacy problems or provide regulators with potential policy tools. Instead, in many cases they are the cause of privacy problems in the first place. This is the case because form and function often reflect different underlying characteristics of an object, in a way that isn't true for the other dimensions.

Focusing on process versus result, or plan versus construction, usually reflects different perspectives or descriptions of an object, but not fundamental substantive differences. With process versus result, for instance, the design process precedes the design result, but both have closely related effects on privacy. Privacy by design is a good example. The basic premise of the movement is that changes to the design process that emphasize privacy will lead to changes in the design result that reflect that emphasis. If privacy is considered from the beginning, then a designer is more likely to make choices that reflect privacy concerns. And this is true whatever the domain: someone designing a software system might choose a database that uses strong encryption by default, while someone designing a ground-floor medical office might choose frosted glass windows to let in light while shielding patients from pedestrians. But these same results can be obtained by regulating the end result—maybe less efficiently, if it means redoing work that was done without privacy in mind, but to the same end. Because of this connection between process and result, there frequently are ways of achieving the same privacy ends by regulating either process or result. And the same is true for planning versus construction.

But form versus function is different. Form and function reflect fundamentally different qualities of an object, so regulations or constraints that affect one may have no parallel with the other. Rules requiring encryption, for instance, necessarily affect the functional design of a product but have no direct or indirect effect on the formal design. Unlike a focus on process versus result or planning versus construction, where a rule about one is often intended to affect the other, there is often little reason to expect that rules about form will affect function or vice-versa.

This disconnect leads to privacy problems because it gives creators a mechanism for exploiting asymmetries between form and function. The user of an object principally experiences its formal design, while the effects of that object on the

world is dictated principally by its functional design. So a creator can develop a form that suggests that it protects privacy, while the function does anything but. And this is especially easy with software, which almost completely severs the connection between form and function in a way that is frequently impossible in the physical world.

A. Form versus Function in the Physical World

In the physical world, form and function are necessarily linked. This is not because, as the cliché alleges, form always follow function. But there is an often-complicated relationship between the two: form is inherently limited and informed by function, and forms can be manipulated to accomplish functions. And because people have been taught by centuries of experience that these connections exist, they make assumptions about function based on the forms of objects.

There are countless examples of the ways that form is limited and informed by function. I previously mentioned the padlock—which is heavy and thick and made of hard metal, and experienced by users as such, because those qualities are necessary to accomplish its function—but the point applies across the physical world. An analog clock is round because its mechanism works by using gears to turn hands around a central axis. Paper is rectangular because larger sheets or rolls can easily be cut into that shape without waste and because it can be fed through a printer without mechanical difficulty. Bottles holding olive oil or red wine are made of brown or green glass because it blocks light, helping the contents last longer. CRT televisions were a foot or two deep because the depth was needed to hold a large cathode ray tube; LCD and plasma televisions aren't because they need no such tube. Forks have pointy bits because tines are the mechanism used by forks to pick up food. Newspaper printing presses are large and complicated and loud because they perform a precise sequence of many steps—printing dozens of pages on rolls of newsprint and collating, cutting, and folding them to produce finished newspapers—at scale, producing thousands of copies an hour. And so forth.

This link between form and function is most strongly associated with modernism and its rejection of ornamentation and embrace of raw materials and pure forms. By reducing an object until it satisfies the minimum requirements to be that object, with only to the components needed to accomplish its purpose, the theory goes, a designer can discover the essential form of that object. The iconic

Parsons table, for instance—a rectangular table with four legs, flush with the corners, square in cross section and the same width as the tabletop—originated out of a challenge to design a table "so basic that it would retain its integrity whether sheathed in gold leaf, mica, parchment, split straw or painted burlap, or even left robustly unvarnished."[54] What are the fundamental requirements of such a table? It needs a surface to put things on and legs or some other kind of support to keep the surface at the right height. And that support should provide stability so the surface doesn't fall over, which suggests placing legs as far from the center as possible. And if you're making the table from raw materials, it's easier if the same materials can be used for the surface and the supports. So from these functional constraints emerges the form of the Parsons table.

The principle, though, that form and function are linked can be overstated. The maxim "form follows function" originated as a prescription for the design of skyscrapers at the turn of the twentieth century, when new technologies and social needs meant it was economical to build tall buildings for the first time. Since such buildings had no established body of architectural precedent, what should they look like? The Chicago architect Louis Sullivan's answer was that "*form ever follows function,* and this is the law."[55] His Guaranty Building in Buffalo, completed in the same year he wrote about form and function, embodied this thinking. The building was divided into three vertical zones: a ground-level zone for retail shops and lobbies, a zone of office floors, and a zone at the top for elevator equipment, utilities, and so forth. And these zones were plainly visible from the outside. The ground-level zone featured large display windows with thick terra cotta blocks in between, decorated with geometric patterns reflecting the blockiness of the windows; the office zone featured narrow windows with (many more) thin columns in between, covered in intricate patterns reflecting the smaller, more intricate façade of the zone; and the utility zone contained arches and the smallest windows of all, more portholes than ordinary windows. Through these zones, the outside of the building made clear exactly what was happening inside, and where.

But Sullivan's was a claim about how tall buildings should be designed, and

54. Mitchell Owens, Dying for a Parsons Table, N.Y. Times, https://www.nytimes.com/2006/06/08/garden/08room.html (June 8, 2006).

55. Louis H. Sullivan, The Tall Office Building Artistically Considered, Lippincott's Monthly Mag., Mar. 1896, at 403, 408 (emphasis in original).

there were always alternatives. Indeed, Sullivan's own career went into a tailspin in the 1890s, and though today he is sometimes credited with founding the modernist school of architecture, his style is highly unusual. Even at the time, its influence was limited; two decades after Sullivan wrote about form and function, the gothic revival Tribune Tower and Spanish colonial revival Wrigley Building opened across the street from each other north of the Chicago River. The Tribune Tower and Wrigley Building may not reflect the minimum essential forms needed to embody office towers, but they are successful, even iconic designs along other metrics.

Still, there is something to the connection between form and function beyond the modernist prescription. People may disagree about whether form *should* follow function to the extent Sullivan and others described—sometimes at the expense of other values and influences—but it is inevitable that form *does* follow function to some extent, even if other factors matter too.

This connection between form and function is fundamental to physical objects. Objects are, well, objects—they are made of physical materials, with physical properties like mass and density and hardness and malleability and brittleness, and the properties of an object are largely determined by the properties of its component materials. Objects accomplish their functions largely through the manipulation and interaction of these physical materials, which is why function inherently limits form. Padlocks, for instance, work by (1) providing a shackle—a physical barrier that is difficult to break or bend or cut—that can be used to lock a locker or attach a bicycle to a rack or prevent some other form of access, and (2) providing a locking mechanical mechanism for opening or closing the shackle, typically by moving tumblers or discs or wheels into a specific position to open the lock. So the shackle has to be small enough to lock a locker or bicycle but hard enough to provide security; the locking mechanism has to be strong enough to provide security but manipulable enough to be readily locked and unlocked; and the lock needs some sort of hardened body around the locking mechanism to make it difficult to manipulate or bypass directly. These are the three standard components of all padlocks, dictated by function, and form follows directly from them.

And people learn about these connections through experience in their day-to-day lives. Objects trigger emotional responses in viewers because a form reveals something about what an object does and how it works, even when someone has never seen an object before or has no idea what it is. Consider the following device:

Is it a piece of industrial equipment, a torture device, something else entirely? Does it provoke warm reactions, like seeing a friendly face, or chilly ones, like seeing a snarling dog or a poisonous snake? The answer to the first question is that it's a spike harrow, a farm tool that's dragged behind a horse to smooth and break up soil but otherwise probably has no particular emotional resonance for those who haven't worked on a farm. But without knowing what it is, a lot of things are clear from its form: it's dangerous, it's the kind of device probably used by experts for something serious, it could probably take off a finger if you did something wrong.

Forms convey meaning in different ways—some might have inherent cognitive meanings,[56] while others might have symbolic meanings or meanings in their cultural contexts. Klaus Krippendorff and Reinhart Butter coined the term "product semantics" to refer to the study of these different meanings of forms in their physical, physiological, social, and cultural contexts.[57] These semantic meanings are both influenced by many factors and come in many forms, some of each beyond

56. Cf. Jake Linford, Are Trademarks Ever Fanciful?, 105 Georgetown L.J. 731 (2017) (using linguistic theory, which suggests that certain sounds have inherent meanings to people, to challenge the assumption in trademark law that fanciful marks have no inherent meaning).

57. Klaus Krippendorff & Reinhart Butter, Product Semantics: Exploring the Symbolic Qualities of Form, 3 Innovation 1 (issue 2, spring 1984). Krippendorff later expanded upon the theme in his book *The Semantic Turn: A New Foundation for Design* (2006). (My copy is still in the mail.)

the designer's control. The meaning a user takes from a product, for instance, is driven by the feedback loop of the user's manipulation of the product in its context and the feedback provided by the object—which are, in turn, influenced by the user's cultural background, literacy of use, mental models, and so forth, and by the product's realization, which is in turn driven by the designer, but also by engineers and commercial constraints and other factors. And the feedback provided by the object can travel through various channels, including obvious ones like information displays and audio feedback, but also the product's form, shape, texture, smell, and so forth.

This complexity means two things: the process of inferring function from form is complicated and sometimes unpredictable—but also a process that is manipulable. If designers understand how forms convey information, they can choose those that are most likely to create the intended meanings. The design firm frog and its founder Hartmut Esslinger, for instance, became famous designing products that were intended to provoke specific emotional reactions in their users—Apple's early Macintosh computers being one of the most prominent examples. Esslinger rejected the maxim that form follows function as leading to cold, minimalist designs that didn't grapple with humans' strive for deeper meanings. Instead he asserted that form follows emotion—that the form of an object should be designed to create specific emotional responses in users. And this became frog's credo. But the fact that form is manipulable and can be used to provoke emotional responses is not obviously inconsistent with the view that form follows function; how an object is experienced by people is itself a part of its function. When Apple chose a teardrop shape and translucent turquoise plastic for the original iMac, it wanted a design that would represent a visual break from the boring, business-like personal computers that came before it. By looking friendlier and more like a consumer product than a piece of high technology, the iMac suggested different uses to its users, like music and video editing. That approachability made the iMac a hit among students and home users.

There are reasons, then, to be skeptical of the maxim that form follows function—or at least, of the strong form of the claim, like Sullivan's assertion that "*form ever follows function,* and this is the law." But there is less reason to be skeptical of the weaker version. Form and function are linked: functions often limit form, and forms can be manipulated to accomplish functions. Industrial designers have

worked with these links for centuries, and will continue to do so for centuries more.

B. Form versus Function in the Digital World

In the digital world, this connection between form and function can be much weaker. With software, the form is the user's experience: what one sees on (usually) a screen and how one interacts with it. And function is what the software does. But instead of function being implemented by physical materials, which limit form, it's implemented by bits. So software can do things that are invisible to users, leaving no footprints other than slightly more memory, storage, and processor use.

Just as there are countless examples of form following function in the physical world, there are countless examples of software that severs this link. There are smartphone flashlight apps that ask a phone for access to a user's location information, photos and videos, and call log, and monetize this information for marketing.[58] There are smartphone apps that appear to be calculator apps but contain hidden photo libraries that you can access by performing a specific calculation. There are online ads that secretly run software to mine Bitcoins using users' processors. Mapping apps gather data about traffic conditions from the aggregate movement of users' phones. Phishing emails purport to contain links to log into your bank or email provider or whoever but really contain links to sites that look like that login but really just steal your username and password. E-commerce websites exist to sell products to users, but often only after those users make it through fraud-detection algorithms. Lots of software adapts or changes itself over time, like personalized search engines or facial recognition that adapts as your face changes or software that changes a screen's color temperature to match ambient lighting conditions. Even mainstream tools used by nearly all computer users contain these mismatches. The "properties" fields in a Microsoft Word document, for instance, by default track various facts and statistics about a document, like its creator, creation data, and total editing time. This is accessible to users but obscure, leading to occasional surprise when, for instance, a student turns in a paper late.

To be sure, it is not possible to completely separate form and function, even

58. E.g., Tom Fox-Brewster, Check the Permissions: Android Flashlight Apps Criticised Over Privacy, Guardian, https://www.theguardian.com/technology/2014/oct/03/android-flashlight-apps-permissions-privacy (Oct. 3, 2014); In re Goldenshores Techs., LLC, FTC no. C-4446 (Apr. 9, 2014) (complaint).

with software. How a piece of software is used and what it enables people to do—the user experience or UX design—are critical parts of its functionality. If two pieces of software do the same thing, but one of them is far easier to use, that one is more successful at accomplishing the function of making it possible for *people* do that thing. And many, maybe most, uses of software require some interface with which users interact. These user-facing tools are part of the formal *and* functional designs of a piece of software. The address bar in a web browser, for instance, is used by users to enter URLs and search terms and to indicate to users what website is loaded; it's also key part of the web browser's now-canonical form. Changes to it, like when Google combined the previously separate URL and search boxes when it released Chrome, have both formal and functional effects. It makes the software look different, but it also changes how users use it. Combining the two boxes eliminated the possibility of confusion when users type or paste text into the wrong box, but also made certain edge cases harder for power users.

But when a function *doesn't* require a user-facing interface, developers can go to town. Functionality that is untethered to form is possible with software because the functions of software are implemented through digital processing that manipulates bits rather than through physical objects that manipulate and interact with physical materials. Since bits aren't manifested in physical form, at least at a scale perceptible by humans, the implementation of those functions doesn't constrain form in the same way that function constrains form with physical objects. Instead, with software, form is just another function implemented by manipulating bits: just as code might implement the process of using an algorithm to encrypt or decrypt data, or the process of uploading a user's location data to a server, it also implements the interface experienced and used by a user—the text and images shown on screen, the action that follows when an on-screen button is pressed, and so forth. Whatever its function and form, software is just bits with the same physical form as any other bits—information downloaded from a server or read off a disk, stored on a hard drive or in flash memory, processed in memory and by a computer's CPU.

And just like form in the physical world is manipulable to induce the desired emotional reactions to an object, form in the digital world is manipulable—except without any of the constraints that physical materials impose. Even some of the most basic metaphors of modern computers, like documents and folders and the

desktop and envelope icons representing email, are just those—metaphors—that are intended to help users understand what is happening in the software behind the scenes. But they have never been literally correct, and there is nothing to stop a developer from choosing different ones. As Orin Kerr observed in 2003, these choices can be made to manipulate users:

> [Computer users'] perceptions reflect the fact that software designers often garnish their applications with icons, labels, and graphics to help novices understand and use them—for example, by writing e-mail programs so that e-mail looks and feels like postal mail. These superficialities have no deeper meaning from the external perspective. What matters is the physical network and the technical details of how it works, not the easily manipulated perceptions of Internet users.[59]

This lack of physical constraints on form gives software developers an immense amount of power to shape the perceptions and experiences of users.

The shift from the physical to the digital is important for privacy because many of the most pervasive privacy problems originate in mismatches between form and function. These mismatches are far more common, and far easier to engineer, in the digital world. Tracking cookies on the web are a good example. The form of a website like the New York Times homepage suggests certain things to users. It suggests that the site comes from the New York Times, through its use of both the Times's logo and the nytimes.com domain name. It suggests that the site provides information to users instead of obtaining information from them, since the site lacks any required login and does not, on its face, have any substantial interactive component. (Compare the Facebook homepage, or that of the financial-tracking site Mint, services that are basically useless unless one is logged in.) It suggests that the information it provides consists of news stories, as suggested by the headlines and short summaries and photos. And it suggests that that information will come from a fairly staid, mainstream publication, through its use of understated, mostly-black-and-white graphic design.

Nothing about this form suggests that the site gathers information about users for advertising purposes or serves code from third parties like Google and Amazon. Nothing about this form suggests that the site partners with a network of

59. Orin S. Kerr, The Problem of Perspective in Internet Law, 91 Geo. L.J. 357, 360 (2003).

other sites to build profiles of users. But those are critical pieces of the site's functional design, and of the vast majority of commercial websites.

There are several recurring fact patterns like this, such as software that gathers data about users for targeted advertising or profile building, or gathers aggregate data about use of the software for bug tracking or market research, or uses computational resources that would otherwise go unused. What all these patterns have in common is the software developer can configure software to do something that benefits the developer, at some cost to users, without users knowing. Often these costs are small or even negligible, like when an ad that uses the user's processing power to mine Bitcoins costs fractions of a penny in electricity costs and increased wear and tear. And often these costs are uncertain or hard to value, like while a company gathers information and uses it for targeted advertising, raising questions about the value of users' information and their privacy. But in the aggregate, they can be substantial, and since users often have no idea they happen, developers don't have to worry about users factoring them into their valuation of the product.

To be sure, this is not a phenomenon that is unique to software; surveillance and other unilateral invasions of privacy have long been possible. And not all privacy problems, even with software, stem from this mismatch between form and function. But the ability to gather information without the constraint of physical form makes it much easier to gather information about people and use that information for gain. If a newspaper wanted to track which stories a reader read offline, it would have to attach a camera to the newspaper or pay a tiny constable to hide in the reader's parlor or something equally implausible. If a newspaper wants to do so online, it can just do so.

We should expect this trend to get worse as software supplants hardware. Many of the same functions that were once accomplished by physical objects can now be accomplished by software, or by software coupled with hardware. This is true even of an object as simple as a padlock; there are dozens of "smart" padlocks for sale, with software-controlled locking mechanisms that are activated by Bluetooth signal or fingerprint reader or electronic keypad. And this is not a new trend. The same pattern—an object that works by manually manipulating physical materials is supplanted by one that works by manipulating bits and using them to control the manipulation of physical materials—played out with printing presses and car engines and kitchen appliances and all sorts of other objects. As more physical

objects get replaced, or supplemented, with software, the opportunity to gather information about users will become more and more tempting for their makers.

IV. IMPLICATIONS AND APPLICATIONS

How well does the divergence of form and function enabled by software describe privacy problems seen in the real world, and how should policy makers react to this divergence? This Part addresses those questions by looking at three common patterns of privacy problems created by software design and the role that form and function play in creating those problems. Two of these recurring problems—deceptive designs and unfair designs—fit within the basic categories of trade practices within the FTC's jurisdiction. The third—insecure designs—has come under the FTC's scrutiny in recent years, but presents its own distinct issues.

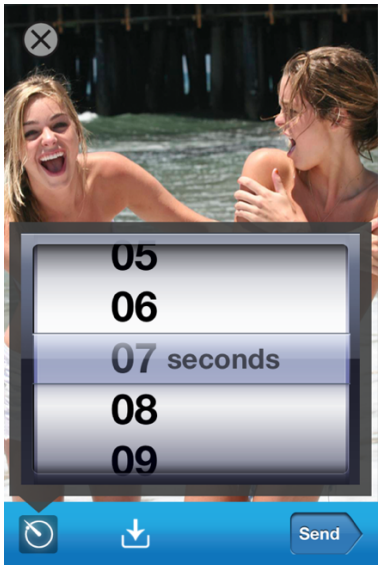
A. Deceptive Design as Form/Function Mismatch

Many privacy problems stem from deceptive designs. These are the most straightforward application of the theory, illustrating the roles that form and function play in privacy problems. But they also show that there exists a class of essentially identical privacy problems that aren't routinely seen as presenting legal problems.

The FTC's settlement with Snapchat over the design of the Snapchat app is a canonical example of deceptive design leading to privacy problems.[60] As Woody Hartzog has explored in detail,[61] the Snapchat app and its marketing conveyed to users—through design, in addition to words—that they could send private photos through the app, which could be seen for a fixed period of time and then would disappear. The app did this by letting users choose a time limit and hit "send" to, well, send the image to someone else; the choice of sample image, cropped to show skin but no clothing, also suggested the kinds of images the app was meant to send:

60. See Snapchat, supra note 42.

61. See Hartzog, supra note 36, at 21–22.



But the impression conveyed by this design (and supported by explicit promises elsewhere) was false; a savvy recipient could save photos they received or view them for longer than the limit. And so, in charging Snapchat, the FTC alleged that Snapchat had committed a deceptive trade practice by "represent[ing], expressly or by implication," that photos would disappear after time expired.[62]

The deceptive Snapchat design was the direct result of a form/function mismatch. The form experienced by Snapchat users strongly implied the app's function. The timer selection tool suggested that users have control over how long someone could see an image, and the choice of sample image suggested that the app was suitable for sending the kinds of sensitive photos that users would want to disappear after the timer was up. But these implications were false. The functional design of the app was dictated by code, which controlled how images were stored, how users could get access to them, how they could be displayed, and so forth. And the formal design of the app was also dictated by code, which created the timer-selection tool, showed the sample image, and so forth. But the two were not linked; nothing would have stopped the app from letting users select a time limit and then just ignoring that selection. Indeed, this is effectively what the app did.

62. See Snapchat, supra note 42, at ¶¶ 6–8, 16–17.

That's a pretty obvious example, but there are many other examples of privacy losses due to deceptive design. A subtler one, for instance, is raised by the Song-Beverly Credit Card Act of 1971, a California statute that bans retailers from asking for "personal identification information" during credit-card transactions. In a pair of cases decided in the 2010s, the California Supreme Court concluded that the Act would be violated when Williams-Sonoma asked a customer for her ZIP code during an in-person transaction,[63] but would not be violated when Apple asked for it during a transaction to buy downloadable content.[64] It's not obvious whether the different outcomes between the two cases are justified; *Pineda v. Williams-Sonoma* was decided unanimously, while *Apple* was a 4–3 decision in which the implications of the shift to online transactions were hotly contested.

The form/function dimension of design might provide some hints about how to best interpret the Song-Beverly Act. The California legislature's concern in enacting the law was that retailers could take advantage of context to falsely imply that providing personal information was necessary. So when Williams-Sonoma asked Pineda for her ZIP code, according to the allegations in her lawsuit, she thought it was needed for the transaction to be processed, but instead Williams-Sonoma used it to build a marketing list. Williams-Sonoma had separated the formal design of its checkout process from its functional design; the process, as experienced by users, was engineered to suggest that ZIP codes were needed to complete the transaction, when the actual function was building a marketing list. (We could think of this as part of the design of the store's point-of-sale software or as part of their standard processes and employee training; whether or not it was embodied in software, this sort of process is like software in that its function mostly isn't implemented by physical materials, and so is unconstrained by physical form.)

A key implication of this view is that it suggests that some designs that have escaped legal scrutiny, at least in the United States, should be examined more closely. Behavioral advertising and cookies used by third-party ad networks to build targeting profiles may be the most significant example. Many academics have long been skeptical of behavioral advertising and related targeting technologies, and in Europe they are often illegal, but United States law tolerates them because they are

63. Pineda v. Williams-Sonoma Stores, Inc., 51 Cal. 4th 524 (2011).

64. Apple Inc. v. Suprerior Court ex rel. Krescent, 56 Cal. 4th 128 (2013).

not "deceptive" or "unfair," as the FTC has interpreted those terms. But the FTC has readily found business practices to be deceptive when the deception was only implicit, based on how a reasonable user would understand a firm's assertions. The form/function mismatch provides a theory of deceptive design that explains when users are likely to make the wrong inferences about how a product works. And these inferences can be engineered. So form/function mismatches are a strong signal that a business practice is deceptive and may merit scrutiny by the FTC.

A second implication is that policies that aim to align form and function could do a better job of providing users with information about companies' data practices, compared to conventional options like disclosure or consent rules. If privacy problems are about notice and an opportunity to make an informed choice—to be sure, a hotly contested issue itself—then finding forms of notice that maximize information and minimize cognitive cost is necessary if people are to make meaningful choices. The form of an object might do a better job of disclosing this information, when the inferences one would draw about the object's function are accurate, because the cost of making such inferences is likely to be much smaller than the cost of reading and understanding a detailed privacy policy. Such policies could take several forms, from something as simple as identifying the biggest problems for the FTC or other regulators to prioritize, to something like Ryan Calo's proposals for "visceral" notice[65] or Paul Ohm's proposal to require companies to change their trademarks when they substantially change their privacy policies.[66]

B. Unfair Design as Form/Function Mismatch

A second group of privacy problems arises due to unfair designs. These are a less-natural fit for the mismatch theory, but they can nevertheless shed light on what might make a design unfair.

The FTC's settlement with FrostWire, which made file-sharing apps for Windows and Android, is maybe the most notable example of unfair software design leading to privacy problems.[67] The FTC asserted that FrostWire's Windows

65. M. Ryan Calo, Against Notice Skepticism in Privacy (and Elsewhere), 87 Notre Dame L. Rev. 1027 (2012).

66. Paul Ohm, Branding Privacy, 97 Minn. L. Rev. 907 (2013).

67. FTC v. FrostWire LLC, No. 11-cv-23643 (S.D. Fla. Oct. 7, 2011) (complaint).

app was deceptive, falsely suggesting that files users downloaded would not be re-shared on the network. But the Android app had no such problem. Instead, the app had an "unfair design," said the FTC, because it made it too hard to limit which files were shared. Users could easily opt to share all of a category of files (e.g., all music files or photos) or none of a category. But if a user wanted to share only certain files, she "first needed to share all the files within the relevant categories and then laboriously unshare individual files in those categories, one at a time, with little or no instruction in the application's user interface about how to accomplish this."[68] Turning off sharing for individual files required selecting a "Browse" icon to see a list of files and then tapping individually on a lock icon next to each file.[69] The meaning of the icon was unexplained, and not especially intuitive, and the process would quickly become unwieldy for users who had a lot of files.

Was there a form/function mismatch in the Android app? In one sense, the two were aligned. The app worked exactly as a user might expect it to, sharing categories of files unless told not to, and withholding exactly the files the user told it not to share. It may not have provided the simplest experience, but few apps do, and that doesn't necessarily indicate that the form would imply something false.

But this account takes too narrow a view of functional design. The function, as FrostWire likely intended it, was to encourage users to share as many files as possible. A file-sharing network only works if it gains a critical mass, so things that help populate the network increase the chance that it will succeed. By setting the default so that whole categories of files were shared and making it difficult to avoid sharing sensitive files, the app's design put a thumb on the scale in favor of sharing. But the app's form did not show the same bias. The app presented itself to users as a way to download share and share those files with others, not a way to share their own personal photos, videos, and other files. It didn't go out of its way to suggest to users that the app would protect their sensitive information, but it also didn't make clear its bias in favor of sharing—a core piece of the app's functional design.

The *Sears Holdings* case is another good example.[70] In that case, Sears offered software that purported to let users join the "My SHC Community," which would

68. Id. at ¶ 25.

69. Id. at ¶ 28–30.

70. In re Sears Holdings Mgmt. Corp., FTC no. C-4264 (June 4, 2009) (complaint).

let them "talk directly to a retailer" and "tell them about the products, services and offers that would really be right for you."[71] Once installed, though, the software's real purpose was to monitor essentially everything a user did online. This was disclosed in a remarkably frank "Privacy Statement and User License Agreement," though the disclosure came 75 lines into the agreement, which was displayed in a scroll box that showed ten lines at a time.

The FTC settled with Sears for allegedly deceptive practices, but the better fit is unfairness. As in *FrostWire,* Sears didn't lie about what it did; it just made it hard to discover a function that was relevant to users deciding whether to join the program. But one way to make something hard to discover and understand—and essentially what Sears did—is to bury it, so the form of the product doesn't reveal its existence. The Sears software did this by highlighting one function and burying another, purporting to be all about joining an online community:

> My SHC Community is a dynamic and highly interactive online community. It's a place where your voice is heard and your opinion matters, and what you want and need counts! As a member of My SHC Community, you'll partner directly with the retail industry. You'll participate in exciting, engaging and on-going interactions – always on your terms and always by your choice. My SHC Community gives you the chance to help shape the future by sharing and receiving information about the products, services and offers that would really be right for you. …
>
> Once you're a member of My SHC Community, you'll regularly interact with My SHC Community members as well as employees of Sears Holdings Corporation through special online engagements, surveys, chats and other fun and informative online techniques. We'll ask you to journal your shopping and purchasing behavior. Again, this will be when you want and how you want to record it—always on your terms and always by your choice. We'll also collect information on your internet usage. Community engagements are always fun and always voluntary![72]

By presenting itself to the user in this way, the software highlighted the function that would appeal to users while slighting the one that would be less appealing.

The punchline of the *Sears* case is that in 2017, eight years after the case was

71. Id. at ¶ 5.

72. Id. at ¶ 6.

settled, Sears petitioned to modify the FTC's order so it could add more tracking features to its mobile apps and weaken the requirement that Sears notify users about tracking features in its apps.[73] Sears argued that the Apple and Google app stores' policies about apps' use of data and the app stores' links to the Sears privacy policy reduced the need for detailed disclosures and FTC-ordered limits on data collection. The FTC agreed, modifying the order in early 2018.[74] But design considerations suggest it was probably wrong to do so. If the takeaway from *Sears Holdings* is that companies can't bury important functions that operate to users' detriment, relying on platforms' policies and disclosures doesn't provide the same benefit since those policies and disclosures are often just as buried.

C. Insecure Design as Form/Function Mismatch

Many of the biggest privacy problems today arise due to security failures. The FTC has gone after companies for data-security failures as both deceptive and unfair trade practices, but security problems are different enough from other kinds of deceptive and unfair practices that they should be considered separately.

Security problems are a puzzle for the mismatch theory. In most cases, security is incidental to the primary functions of a piece of software. When security problems have led to substantial data breaches or legal problems, security hasn't been the principal function of the product, or even one of the functions that would be salient to users; instead, users have turned to the product for something like buying movie tickets[75] or monitoring one's credit information[76] or booking hotel rooms[77] or running medical tests.[78] So people are unlikely to infer much about the functional design of the embedded security system from the product's form.

But the same can be true offline, and security is a context where that is especially likely. Sometimes security is readily apparent from form; consider barbed-

73. In re Sears Holdings Mgmt. Corp., FTC no. C-4264 (Oct. 30, 2017) (petition to reopen and modify final order).

74. In re Sears Holdings Mgmt. Corp., FTC no. C-4264 (Feb. 27, 2018) (order).

75. In re Fandango, LLC, FTC no. C-4481 (Aug. 13, 2014) (complaint).

76. In re Credit Karma, Inc., FTC no. C-4480 (Aug. 13, 2014) (complaint).

77. FTC v. Wyndham Worldwide Corp., 799 F.3d 236 (3d Cir. 2015).

78. LabMD, Inc. v. FTC, 894 F.3d 1221 (11th Cir. 2018).

wire fences or thick bank-vault doors or armed security guards. This is especially common when security is highly salient to people—as in a bank or airport or military facility—both because visibility is part of the functionality of an effective security system and because many stronger security techniques, like thick vault doors, can't effectively be hidden. Other times security is more hidden, like a bank teller's hidden emergency alarm button or a storefront security grille that rolls up into the ceiling while the store is open. This happens when visible security would be counterproductive and when form permits it, but it always involves some compromises, because the security apparatus needs to fit into the limits of the store's formal design. Those compromises might not be as bad as a padlock made out of tissue paper or aluminum foil, but like such a weakened padlock, they represent cases of the needs of form limiting function.

Software is similar. Companies developing apps have an interest in using security techniques that don't interfere with the user experience. This is why companies resist strong password requirements and requiring two-factor authentication and especially stronger forms of two-factor authentication like authenticator apps or hardware authentication devices. A hotel that required every guest to go through a metal detector and bag search would lose business unless it was in an area where such security was understood by customers to be necessary, in which case *not* doing so would cost it business. And an app that required customers to have a 64-character password with hardware authentication would lose business unless it was serving a purpose—probably not buying movie tickets—where such security was seen by users as necessary.

Security, then, is an area where the mismatch theory plays out in some contexts where people can readily make inferences about function from form—the form of a bank vault, or an armed guard—but not in others. In those other cases, people fall back on their baseline assumptions about how the world works. In the physical world, people might assume that a store, or a hotel, or a bank takes reasonable precautions to protect them from risks like robbery and fraud. This assumption is backed up by different mechanisms in different contexts. Sometimes the law imposes fiduciary duties or requires businesses or people to hold insurance or imposes tort liability for failures to protect people; sometimes the marketplace punishes companies that fail to live up to customer expectations.

But the fact that consumers fall back on baseline assumptions doesn't mean

there's no inference being made from form. A critical part of relying on such background assumptions is the *lack* of any indication that such assumptions are wrong. Restaurants work to maintain clean and well-designed front-of-house areas in part because customers like dining in a nice environment, but also because care in the visible parts of the business suggests care in the invisible parts; if the dining room is filthy, the kitchen probably is too. A pawn shop with bars on the windows suggests something different from one with no bars on the windows: that it has better security, but also that it *needs* better security.[79]

A baseline assumption that a company can be trusted to protect customers, then, partly reflects the lack of any obvious warning signs. By that metric, software that seems reasonably directed to normal, unrelated tasks like booking a hotel room or buying a movie ticket should indicate, through the lack of any obvious sign of problems, that the provider is trustworthy to protect the user. When that trust fails, there has been a form/function mismatch: the form implied a general trustworthiness, which the function failed to deliver.

What does this imply for regulation? For one thing, it suggests that the FTC is right that lax data security can harm consumers in ways that can't easily be avoided, and so is likely to be an unfair trade practice. But more interestingly, it suggests that when a company has failed, one potential remedy might be to require their software to embrace forms that suggest a lack of trustworthiness. That might mean something like a pop-up warning when a user launches an app, or a red warning banner on a website, or something like requiring companies that have experienced data breaches to show photos of people affected by the breach. Such disclosures are likely to be far more visceral and salient to users than the fine-print disclosures required by state data-breach-notification laws.

CONCLUSION

And now the paper is over. I hope you enjoyed it. Wouldn't it be nice if all papers ended this way?

79. Cf. Kyle Bagwell, The Economic Analysis of Advertising, in 3 Handbook of Industrial Organization 1701, 1716–20 (Mark Armstrong & Rob Porter eds., 2007) (exploring different ways that advertising can provide information, including by the very act of advertising itself, which suggests that a company is healthy enough to advertise in the first place).