Privacy Risk Analysis Based on System Control Structures

Adapting System-Theoretic Process Analysis for Privacy Engineering

Stuart S. Shapiro
The MITRE Corporation
Bedford, MA USA
sshapiro@mitre.org

***Abstract*—To date, top-down efforts to evolve and structure privacy engineering knowledge have tended to reflect common systems engineering/development life cycle activities. A different approach suggests a particular need for technical analytical methods. To help address this need, this paper proposes to adapt for privacy engineering an existing technique, System-Theoretic Process Analysis (STPA), developed for safety engineering. The foundations of STPA are discussed, its security extension, STPA-Sec, is described, and modifications to STPA-Sec are proposed to produce STPA-Priv. STPA-Priv is then applied to a simple illustrative example.**

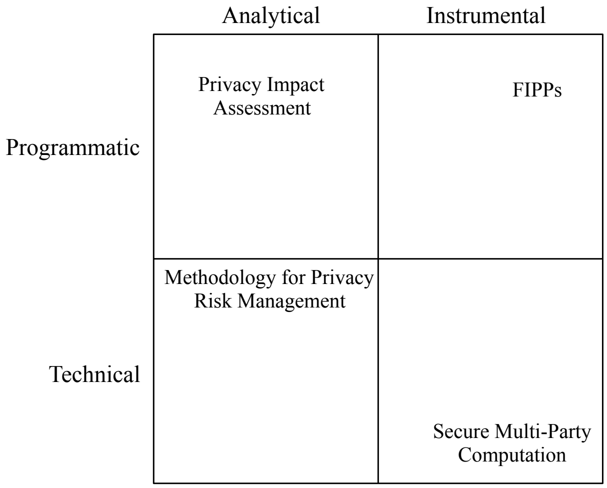
Keywords—privacy risk analysis; System-Theoretic Process Analysis; STPA; STPA-Sec; STPA-Priv

I. Introduction

To date, top-down efforts to evolve and structure privacy engineering knowledge have tended to reflect common systems engineering/development life cycle activities. (In contrast with more bottom-up efforts, such as [1].) While useful, an unfortunate side effect of such approaches is a tendency to focus on existing development techniques while folding in privacy concerns. The process described in [2] is one example of this. Somewhat broader, but similarly dominated by conventional techniques, is the process described in [3]. While, on the one hand, the OASIS Privacy Management Reference Model and Methodology (PMRM) [4] breaks more new ground, on the other hand, it is a comprehensive approach that is largely all or nothing. (In effect, the life cycle becomes the methodology.) The EU PRIPARE Privacy and Security-by-design Methodology [5] is perhaps the most catholic of any of these attempts, referencing or incorporating a number of privacy-specific as well as more general techniques, including the PMRM.

A coarser structure may provide a more flexible starting point for organizing existing privacy engineering techniques (which can then be mapped to specific life cycles) and driving the development of new privacy-specific ones. One approach is to use a quadrant chart with one axis differentiating between programmatic and technical methods and the other axis differentiating between analytical and instrumental methods. Programmatic methods (e.g., life cycles) are those that focus on organizational processes and activities, while technical methods (e.g., formal specification) are those that involve application of specialized knowledge. Analytical methods (e.g., risk assessment) are those that support examination, while instrumental methods (e.g., design patterns) are those that support creation. (It should be noted that some methods can serve both analytical and instrumental functions, depending on whether they are applied to something that already exists or something that is being created.) Privacy engineering, like any other engineering discipline, requires substantive methods residing in all four quadrants, as illustrated (with a few examples) in Fig. 1.

Indeed, the history of privacy as an interdisciplinary field can be viewed as the gradual population of these quadrants over time. Programmatic instrumental elements arose first, driven by statutes, regulations, policies and procedures (much of them grounded in some form of Fair Information Practice Principles (FIPPs)), and this continues to be the dominant quadrant. This was followed by development of some programmatic analytical techniques in the form of privacy impact assessments (PIAs) and, later, by more sophisticated approaches such as contextual integrity [6]. Meanwhile, the growth of privacy enhancing technologies (PETs), including secure multi-party computation [e.g., 7] and differential privacy [8], has provided some technical instrumental methods,



	Analytical	Instrumental
Programmatic	Privacy Impact Assessment	FIPPs
Technical	Methodology for Privacy Risk Management	Secure Multi-Party Computation

Fig. 1. Engineering Knowledge Quadrants with Examples of Methods

Portions of this material were presented in a talk at the 2016 MIT STAMP Workshop. Approved for Public Release; Distribution Unlimited. Case Number 16-0355

among other privacy-specific methods, in addition to more general existing methods such as process modeling.

While much remains to be done to build out these three categories of methods to the point at which they can robustly support privacy engineering, lagging behind all of them is technical analytical methods. While some modest headway has been made in technical methods for privacy risk assessment, for example, a 2013 report on privacy risk management for the UK Information Commissioner's Office [9] identified only three privacy-specific approaches. Of these, only [10] really qualifies as a technique in and of itself and even it arguably sits on the border with programmatic methods.

This paper aims to contribute to the development of technical analytical methods, specifically risk analysis methods, for privacy engineering by adapting an innovative method developed to perform risk analyses for safety engineering of socio-technical systems: System-Theoretic Process Analysis (STPA) [11]. STPA analyzes system control structures for hazards that, under certain causal scenarios, could result in system behavior that violates safety constraints. A modified version of STPA, STPA-Sec [12], has been put forward to support risk analyses for security engineering. This paper proposes to further modify STPA-Sec to support technical risk analysis for privacy engineering, yielding STPA-Priv.

The remainder of the paper is organized as follows. Section II provides background on STPA while Section III describes STPA-Sec. Section IV describes how STPA-Sec might be extended to address privacy, becoming STPA-Priv. Section V applies STPA-Priv to the simple illustrative example of a smart television and relates STPA-Priv to some other analytical methods. Finally, Section VI discusses future work to evolve STPA-Priv into a generally accessible privacy engineering technique.

II. STPA Origins

STPA is rooted in the application of systems theory to safety. More specifically, it is grounded in System-Theoretic Accident Model and Processes (STAMP) [11], an accident causality model developed in response to the inadequacies of traditional accident models when analyzing complex socio-technical systems. While [13] highlighted these kinds of problems some time ago, the issues posed by modern socio-technical systems go beyond higher complexity and tighter coupling. Emergent properties, use of digital rather than analog technology, increases in accident impact breadth and depth, trade-off interdependencies, and human cognitive limitations have also undermined the utility of traditional safety engineering techniques.

STAMP frames safety in terms of constraints rather than events. Safety is achieved through the proper enforcement of complete and correct constraints on system behavior, rather than the prevention of certain events or chains of events. The more inclusive notion of behavioral constraints has the potential to identify problems arising out of issues such as unanticipated component interactions. Constraints are enforced through controls, which can be either passive or active. Passive controls enforce constraints simply by their presence. An electrical fuse is an example of a passive control; if the fuse "blows" (excessive electrical current melts the conductor), the circuit is broken. Active controls, in contrast, achieve their effect through some deliberate action. If, instead of using a fuse, a computer monitored the electrical current and broke the circuit if it detected excessive current, this would constitute an active control.

Hierarchical control structure is another fundamental concept in STAMP. Controls operate between hierarchical levels with controls at each level imposing constraints on processes in the level below it. This control structure exhibits multiple aspects. As described by Leveson [11], controls invariably involve adaptive feedback mechanisms, i.e., they are closed-loop controls. Communication channels carry control commands to the relevant processes and information from the processes to the controllers. While the focus in STAMP is on feedback loops, though, it is also possible for controls to operate in the absence of feedback, i.e., as open-loop controls. This has implications for applying STPA to privacy.

Another key aspect of hierarchical control structure in STAMP is that it applies to both the development and operation of socio-technical systems. Systems engineering/development life cycles utilize controls just as much as the systems they produce do, and these controls affect system safety and other properties just as much as the controls that apply to system operation. This highlights the fact that we are dealing with socio-technical systems and that controls will be social (broadly construed) as well as technical. Thus, for example, the highest hierarchical level of the system may well be government, as it is government that passes statutes and implements them via regulation. Given the historically fundamental roles played by statutes and regulations in privacy, explicit accommodation of these kinds of controls is crucial if STAMP is to serve as the foundation of a form of STPA that can be applied to privacy.

One more principal concept in STAMP is process models. For a control to properly constrain a process, it must maintain a model of that process. Accidents can occur when the process model being used by the controller diverges from the actual process being controlled. This can result in four different types of control errors:

- Incorrect control action
- Missing control action
- Control action provided at the wrong time
- Incorrect duration of control action

Note that these control errors are derived from the more general systems theory that informs STAMP. Control errors arise when the process model being used by a controller doesn't properly correspond to the process being controlled. (This can happen either because there is an error or gap in the model or because its state does not match the actual process state.) Therefore, there is no prima facie reason to think that this typology is domain specific and would not apply to privacy.

STAMP forces reconsideration of a number of the underlying assumptions of traditional safety engineering. In particular, for our purposes, STAMP questions the suitability of probabilistic risk analysis. The difficulty of applying probabilistic risk analysis to privacy (and security), therefore, may reflect an intrinsic problem rather than an immaturity of technique. If so, an adaptation of STPA, a method based on STAMP, for privacy engineering may prove a valuable alternative to a number of the current approaches to privacy risk analysis, such as [10], that take probabilistic risk analysis as their inspiration.

III. STPA and STPA-Sec

STPA aims to operationalize the insights of STAMP and apply them to safety engineering of modern socio-technical systems. By virtue of this, STPA can identify and address causal factors that traditional safety engineering techniques do not easily accommodate. STPA accomplishes this by systematically structuring the identification of constraints and the controls that enforce them and the conditions under which control errors may lead to constraint violation and, thus, an unsafe system state.

STAMP frames safety as a product of appropriately constrained system behavior. Security can be similarly framed. Indeed, security engineering labors under the same kinds of problems that motivated the development of STAMP and STPA for safety engineering. This observation led to the extension of STPA to security through STPA-Sec. Because speaking in terms of STPA-Sec moves the discussion into a more salient domain than safety, we will describe STPA-Sec rather than STPA, enabling the use of language more resonant with privacy as well as security practitioners.

STPA-Sec involves four principal steps [12], as detailed below.

A. *Identify losses to be considered*

As a first step toward identifying necessary constraints on system behavior, the outcomes to be avoided have to be articulated. STPA refers to losses, and so does STPA-Sec. Losses due to insecurity can include human injury, physical damage, degraded reputation, and economic loss, among others. In cyber security terms, these map to varying extents to the objectives, and potential loss, of confidentiality, integrity, and availability (C-I-A). One can decompose these further into lower level losses, but the marginal return on such decomposition can quickly decrease.

B. *Identify system vulnerabilities that can lead to losses*

System vulnerabilities (hazards in STPA) are potential system states that, together with worst-case environmental conditions, may result in a loss. In practice, they can be thought of as a form of anti-goal [14], those high level states that, if not avoided, could lead directly to a loss. These form the basis of relevant constraints on system behavior, which are established in the next step.

C. *Specify system functional control structure*

Security constraints that the system must enforce are derived from the system vulnerabilities. These are effectively the converse of the vulnerabilities. If the system is being developed, constraints that address the identified system vulnerabilities must be defined and implemented via specification of the system's functional control structure. If an existing system is undergoing analysis, and the functional control structure has not already been documented, it must be captured from the system design. Functional control structures typically are represented using block diagrams depicting the relevant entities (including individuals and organizations as well as technical components) along with the control and feedback flows. The granularity of the representation typically will be driven by the vulnerabilities. In other words, the representation must be at a level of detail such that it conveys controls that address the identified vulnerabilities.

This will entail varying degrees of technical detail depending on the system. As with the identification of losses, the functional control structure can be decomposed to whatever extent is necessary. In an STPA case study involving a spacecraft, for example, a Level-0 diagram for a particular operation and a Level-1 diagram of one subsystem were generated [15]. (There are as yet few publicly available STPA-Sec case studies, so we reference STPA case studies for illustrative purposes.)

D. *Identify insecure control actions*

Identification of insecure control actions can be driven by filling in a table in which the columns correspond to the four types of control action errors implied by STAMP. The rows correspond to the security constraints. Each cell may contain multiple entries where a constraint may be violated by different control action errors of the same type. Causal scenarios are then generated to reveal specific situations which could prompt the erroneous control action. These scenarios include ones involving intentional actions by threat actors, unlike STPA which only concerns itself with unintentional actions.

Causal scenarios typically exhibit the greatest amount of technical detail since they describe the specific conditions under which an erroneous control action might occur. The control action analysis identifies *what* might go wrong while the causal scenarios identify *how*. In an application of STPA to an avionics system, for example, the control structure was represented at the level of flaps, levers, hydraulic lines, sensors, and displays. However, the causal scenarios were described in terms of specific flap positions ("detents"), sensor readings, and system messages, among other details [16]. A decision must then be made regarding whether to mitigate the risk represented by the scenario and, if so, how.

IV. Adapting STPA-Sec for Privacy

STPA-Priv extends STPA-Sec to address privacy in two principal ways. One involves defining losses while the other involves capturing the control structure. Neither of these changes the nature of STPA-Sec or the steps required to carry it out. Rather, in keeping with the notion of extension, they bring in additional concepts.

STPA focuses on potential losses of concern, which is quite sensible in a safety context. STPA-Sec also emphasizes losses, and this works as well, especially when thinking in terms of cyber security and the classic C-I-A triad. A focus on loss, while not incompatible with STPA-Priv, works less well in the privacy domain, in part because there is no agreed set of objectives corresponding to C-I-A. (The U.S. National Institute of Standards and Technology (NIST) has proposed some—predictability, manageability, and disassociability [17]—but how much traction these will gain remains to be seen.) While in some respects privacy can be straightforwardly conceptualized in terms of loss—e.g., loss of confidentiality, loss of contextual integrity—in other cases it becomes more awkward, particularly when invoking harms such as Solove's taxonomy [18] or violations of FIPPs. More workable for privacy than the concept of "loss" is the notion of "adverse consequence," recognizing that, unlike the universally embraced C-I-A in security, there are a variety of approaches to framing privacy and adverse privacy consequences, including those above as well as others, such as LINDDUN [19]. Whichever specific privacy framework is chosen, it is this framework that, either directly or inversely, defines privacy and adverse privacy consequences for the purpose of a given use of STPA-Priv.

When describing control structures, both STPA and STPA-Sec focus on closed-loop controls that include feedback mechanisms, enabling adaptive control. While it is true that systems theory emphasizes closed-loop controls, especially for open systems (i.e., systems that interact with their environment), open-loop (i.e., non-adaptive) controls are also recognized in systems theory and can be, for better or worse, used in open systems [20]. Arguably, some foundational privacy controls can act as open-loop controls, including various forms of notice and consent. Individuals may or may not read a privacy notice, but there is often no feedback to the control. A similar argument can be made for implicit consent. Other controls, such as differential privacy, are more typically closed-loop controls, as epitomized by the real-time adjustments that a differentially private system will make in response to queries.

This raises the question of whether STPA-Sec and/or STPA themselves would benefit from explicitly accommodating open-loop controls as well. An answer to this question, though, is outside the scope of this paper. Irrespective of whether such a case exists for security, we can demonstrate by example that such a case exists for privacy.

Extending STPA-Sec to become STPA-Priv, therefore, requires explicit accommodation of open-loop controls and different frameworks for adverse consequences. This results in the following steps for STPA-Priv:

1. Identify potential adverse privacy consequences to be considered, as denoted by a selected framework
2. Identify vulnerabilities that can lead to adverse privacy consequences in the context of the system
3. Specify system privacy constraints and functional control structure, including open-loop privacy controls
4. Identify privacy-compromising control actions

In the next section, we will illustrate these steps using the simple example of a smart television.

V. Applying STPA-Priv

This example is based on an actual smart television from a major manufacturer. For the purposes here, we focus on the manufacturer's privacy policy and the feature that enables the television to recognize on-screen content. Lacking further specifics, we have inferred or postulated details where necessary. Therefore, the following analysis reflects the general lack of available technical specifics and is not necessarily complete or accurate.

The feature is enabled by default, though it can be turned off by the user. After the feature has been disabled, though, any data previously collected will continue to be used for an indeterminate period of time. When enabled, the manufacturer will collect data related to publicly available displayed content, such as the service provider and the time, date and channel of programs and commercials viewed ("viewing data"). Viewing data are not collected from televisions located outside the United States.

Viewing data are claimed to be anonymous and are combined with IP address (also claimed to be anonymous) and other information, such as demographic information obtained from third parties, to guide selection and delivery of ads by third parties and to further analyze the data. These ads may be delivered to other Internet-connected devices of which third parties are aware that share the television's IP address. Aggregate viewing data are shared with media and data analytics companies. In most cases, IP addresses are hashed. Conditions of confidentiality and use apply to the sharing of unhashed IP addresses with third parties. Third parties receiving viewing data are also required to employ reasonable security measures. Viewing data are encrypted before being transmitted over the Internet. The analysis proceeds through the four steps as follows.

A. *Step 1: Identify potential adverse privacy consequences to be considered, as denoted by a selected framework*

For simplicity, we will utilize Calo's subjective/objective privacy harms [21]. This constitutes the privacy framework we will employ to identify adverse privacy consequences. A subjective privacy harm is the perception of unwanted surveillance. An objective privacy harm is the forced or unanticipated use of personal (i.e., specifically related to a person) information. In the next step, we will consider these consequences in the context of the smart television example described above.

Note, though, that using a relatively coarse privacy framework means that much of the analysis will be equally coarse. Further, owing to the general lack of available technical specifics, this coarseness will carry over into the casual scenarios as well. The granularity of the example, therefore, is less reflective of the method than of the availability of system information and the choice of privacy framework. More detailed system information, such as actual design specifications, and a more granular privacy framework, such as LINDDUN, would produce a more technical analysis.

B. *Step 2: Identify vulnerabilities that can lead to adverse privacy consequences in the context of the system*

We then combine the identified adverse consequences with the context of the system to identify system vulnerabilities, i.e., system and environmental states that may lead to an adverse privacy consequence. In other words, we aim to identify those situations grounded in the characteristics of the system and its environment which could result in a subjective or objective privacy harm:

- User of device associated with the same IP address as the television may perceive unwanted surveillance based on the ads delivered, even if not responsible for program choices.
- User does not realize prior to use how viewing data are being collected, retained, combined with other information, and used to serve ads and for other analytics.
- User wants to opt out of collection of viewing data but cannot determine how to disable collection.

C. *Step 3: Specify system privacy constraints and functional control structure, including open-loop privacy controls*

The vulnerabilities identified in the previous step can be reframed as a set of privacy constraints that the system must enforce, namely:

- User of device associated with the same IP address as the television must not perceive unwanted surveillance based on the ads delivered.
- User must understand what and how data are being collected and used and actual practices must be consistent with that understanding.
- User must be able to determine how to disable collection of viewing data and to carry out those instructions.

Fig. 2 shows the high-level functional control structure for the system. While the boundary of a socio-technical system is essentially arbitrary, for this kind of analysis it must be drawn in a way that captures relevant context but minimizes the components over which the privacy [and other] engineers have no control. Note that because the diagram represents control structure rather than data flows, data are only included when they are germane to functional control. We enclose the term anonymization in quotation marks to acknowledge its contested nature, which is further called into question by the inclusion of IP addresses.

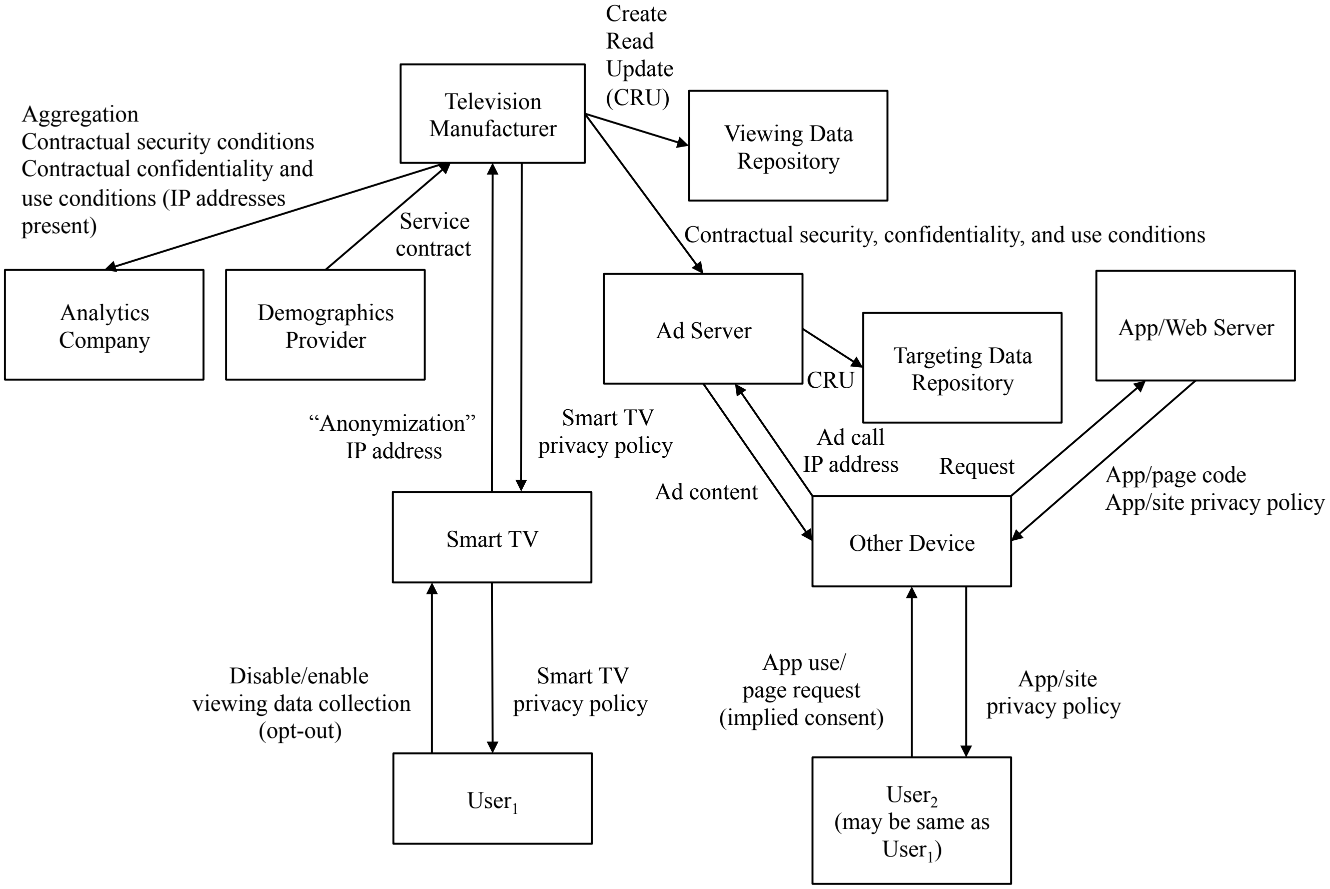


Fig. 2. Functional Control Structure For Smart TV Feature

D. *Step 4: Identify privacy-compromising control actions*

Two steps are required to identify actual privacy risks. Table I captures the first: the control action analysis. For each privacy constraint, erroneous control actions that could violate that constraint are identified. Note that an erroneous control action may apply to more than one constraint. The analysis identified ten unique erroneous control actions.

It can be useful when performing this analysis to explicitly consider the process model associated with each control and the ways in which the model might diverge from the actual process state. This could result, for example, from inaccurate feedback and manifest as a failure to recognize that a television is outside the U.S. One can pursue this more rigorously by explicitly considering key process variables or, if available, an applicable state diagram.

In the second step, based on the control action analysis, causal scenarios (reflecting worst case environmental conditions) are developed. These are listed in Table II. (Each erroneous control action is listed once, eliminating duplicates.) Only erroneous control actions for which a causal scenario can be described constitute actual risks. In this case, at least one causal scenario could be described for each unique erroneous control action.

The causal scenarios suggest potential changes to the control structure to mitigate the risk of system behavior that violates privacy constraints. For example, switching to opt-in consent for the collection of viewing data by disabling the feature by default would partially address several problematic control actions. Determining appropriate responses to identified risks is, strictly speaking, outside the scope of STPA-Priv as an analytical technique (and is similarly outside the scope of STPA and STPA-Sec), though any changes effected could be fed back into the analysis.

E. *Relation to Some Other Methods*

STPA-Priv, as well as STPA-Sec and STPA, bear some relationship to goal-oriented modeling [14], since they deal with different types of non-functional requirements implicitly in terms of goals and anti-goals. However, risk analysis in goal-oriented modeling bears some resemblance to fault tree and similar forms of hazard analysis. Since the fundamental motivation behind STPA is the perceived inadequacies of such analytical techniques, one can hypothesize that risk analysis techniques grounded in goal-oriented modeling may suffer similar problems. Section VI discusses possible future work that could test this hypothesis.

Irrespective of that hypothesis, STPA-Priv as an analytical technique offers the benefit of applicabilty to an existing system design irrespective of what techniques were employed to develop it, including the particular life cycle used. (The use of STPA as an instrumental method is treated separately as “safety-guided design” [11].) Further, STPA-Priv strikes a balance between prescription of privacy framework (e.g., LINDDUN [19]) and completely open-ended goal-oriented modeling (e.g., KAOS [14]), enabling use of any defined privacy framework as a basis for analysis.

TABLE I. CONTROL ACTION ANALYSIS FOR SMART TV FEATURE

Privacy Constraint	Incorrect control action	Control action not provided	Control action provided too soon or too late	Control action applied too long or not long enough
User of device associated with the same IP address as the television must not perceive unwanted surveillance based on the ads delivered.	Transmission of viewing data from TV outside the U.S. enabled	Privacy information not provided to user in the context of the device User is not empowered to disable collection of viewing data		
User must understand what and how data are being collected and used and actual practices must be consistent with that understanding.	Transmission of viewing data from TV outside the U.S. enabled Privacy information unclear Micro-level data can be inferred from aggregate data Micro-level data can be associated with identifying information	Privacy information not read Data are not deleted or are deleted inconsistently from the viewing and targeting data repositories	Privacy information not communicated prior to TV use	
User must be able to determine how to disable collection of viewing data and to carry out those instructions.	Instructions and/or control for disabling collection of viewing data not readily accessible	User is not empowered to disable collection of viewing data		

TABLE II. CAUSAL SCENARIO GENERATION

Problematic Control Action	Causal Scenarios
Transmission of viewing data from TV outside the U.S. enabled	VPN use results in TV outside the U.S. being associated with a U.S. IP address
Privacy information not provided to user in the context of the device	User of device has not reviewed privacy policy on TV and experiences ads that appear to reflect viewing habits
User is not empowered to disable collection of viewing data	User makes use of the TV but does not have the authority to disable collection of viewing data due to their position or role (e.g., a child or visitor in a home)
Privacy information unclear	Privacy policy provides information that is too general or too detailed to understand Privacy policy is poorly written for a general reader
Micro-level data can be inferred from aggregate data	Data are aggregated in such a way as to enable data associated with specific smart TVs to be recovered by analytics firms
Micro-level data can be associated with identifying information	As multiple sets of "anonymous" data are combined, it becomes possible to link data to specific individuals or households via quasi-identifiers
Privacy information not read	User ignores privacy policy when presented
Data are not deleted or are deleted inconsistently from the viewing and targeting data repositories	No explicit retention policy exists for data in the viewing and targeting data repositories; retention policy is implicit based on how information categories are defined in the privacy policy
Privacy information not communicated prior to TV use	Privacy policy is not presented to all individual users upon initial use
Instructions and/or control for disabling collection of viewing data not readily accessible	User can't find or can't remember where to find instructions and/or control for disabling collection of viewing data User has difficulty following instructions for disabling collection of viewing data

VI. Conclusion

The extent to which STPA has successfully identified safety risks missed by traditional techniques [e.g., 22] lends hope that STPA-Priv might do the same for privacy. It offers potential benefits when dealing with more complex systems by forcing systematic analysis of system controls and their ability to constrain behaviors that might compromise privacy. It should be noted that the back half of the process—capturing the functional control structure and analyzing controls—is not strictly linear and is more a matter of iterative refinement. Working through any one of the control structure, control action analysis, and causal scenario generation will prompt changes to the others. The ultimate result, then, will be more a matter of convergence than of reaching the end of a straightforward linear process. Such a process, arguably, is more likely to successfully accommodate the characteristics of complex socio-technical systems.

Work to further develop STPA-Priv will encompass three stages. The first stage will involve refining the method as described above. The second stage will involve documenting the refined method in a manner that effectively supports operationalization for both systems engineering/development processes and the systems themselves. The final stage will involve applying the documented method to a real-world project with privacy implications to initially gauge its practicality.

Ideally, the utility of STPA-Priv would then be further validated through a controlled experiment. This would involve two independent teams performing a privacy risk analysis on a relatively complex system. One team would employ STPA-Priv while the other would employ another, existing method (such as [10] or that described in [14]). Both effort and results would be compared and appropriate conclusions drawn regarding relative efficiency and efficacy.

A less controlled, but more practical experiment, similar to [19] for STPA, would analyze a system for which a privacy risk analysis had already been performed and documented, assuming nobody on the STPA-Priv team had seen the other analysis. If STPA-Priv identified privacy risks missed by the other method, this would instill greater confidence in its value as a stand-alone technique. If each method identified risks that the other method missed, this would imply that the value of STPA-Priv might be as a complement to other privacy risk analysis methods. If STPA-Priv produced results that were no better, but no worse, than the other method, this would not invalidate it as it could still serve as an alternative technique based on individual or team preference. If, however, STPA-Priv identified only a subset of the privacy risks identified by the other method, this would cast serious doubt on its utility.

Acknowledgment

Thanks to MITRE colleague Julie Snyder for comments on an earlier version of this paper.

References

[1] Computing Community Consortium Privacy by Design Visioning Activity, http://cra.org/ccc/visioning/visioning-activities/2015-activities/privacy-by-design/ (accessed 28 January 2016).

[2] M. F. Dennedy, J. Fox, and T. R. Finneran, The Privacy Engineer's Manifesto: Getting from Policy to Code to QA to Value. New York: Apress, 2014.

[3] I. Oliver, Privacy Engineering: A Dataflow and Ontological Approach. n.p.: CreateSpace Publishing, 2014.

[4] Organization for the Advancement of Structured Information Standards (OASIS), Privacy Management Reference Model and Methodology (PMRM) Version 1.0, Committee Specification 01, 3 July 2013. Burlington, MA, 2013.

[5] PReparing Industry to Privacy-by-design by supporting its Application in Research (PRIPARE), Privacy and Security-by-design Methodology (Deliverable D1.2, 12 November 2014). Paris, 2014.

[6] H. Nissenbaum, Privacy in Context: Technology, Policy, and the Integrity of Social Life. Palo Alto: Stanford Law Books, 2009.

[7] C. Orlandi, "Is multiparty computation any good in practice?," Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pp. 5848-5851, 22-27 May 2011.

[8] C. Dwork, "A firm foundation for private data analysis," Commun. ACM, vol. 54, pp. 86-95, January 2011.

[9] Trilateral Research & Consulting, Privacy impact assessment and risk management, Report for the Information Commissioner's Office, 4 May 2013. Wilmslow, Cheshire: UK Information Commissioner's Office, 2013.

[10] Commission Nationale de l'Informatique et des Libertés (CNIL), Methodology for Privacy Risk Management (translation of the June 2012 edition). Paris, 2012.

[11] N. G. Leveson, Engineering a Safer World: Systems Thinking Applied to Safety. Cambridge, MA: MIT Press, 2011.

[12] W. Young and N. G. Leveson, "An integrated approach to safety and security based on systems theory," Commun. ACM, vol. 57, pp. 31-35, February 2014.

[13] C. Perrow, Normal Accidents: Living with High-Risk Technologies. Princeton, NJ: Princeton University Press, 1999.

[14] A. van Lamsweerde, Requirements Engineering: From System Goals to UML Models to Software Specifications. Chichester, UK: Wiley, 2009.

[15] T. Ishimatsu et al., "Hazard analysis of complex spacecraft using Systems-Theoretic Process Analysis," AIAA Journal of Spacecraft and Rockets, vol. 51, pp. 509-522, March 2014.

[16] C. H. Fleming and N. G. Leveson, "Improving hazard analysis and certification of integrated modular avionics," Journal of Aerospace Information Systems, vol. 11, pp. 397-411, June 2014.

[17] U.S. National Institute of Standards and Technology (NIST), Privacy Risk Management for Federal Information Systems, NISTIR 8062 (Draft, May 2015). Gaithersburg, MD, 2015.

[18] D. Solove, Understanding Privacy. Cambridge: Harvard University Press, 2010.

[19] M. Deng, K. Wuyts, R. Scandariato, B. Preneel, and W. Joosen, "A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements," Requirements Engineering, vol. 16, pp. 3–32, March 2011.

[20] D. H. Meadows, Thinking in Systems: A Primer, ed. Diana Wright. White River Junction, VT: Chelsea Green, 2008.

[21] M. R. Calo, "The boundaries of privacy harm," Indiana Law Journal, vol. 86, pp. 1131-1162, 2011.

[22] C. H. Fleming, M. Spencer, J. Thomas, N. Leveson, and C. Wilkinson, "Safety assurance in NextGen and complex transportation systems," Safety Science, vol. 55, pp. 173–187, 2013.