CHAPTER

5

Deconstructing security and privacy issues: The development of a logic for capturing mismorphisms*

Vijay H. Kothari, Prashant Anantharaman, and Sean W. Smith

Department of Computer Science, Dartmouth College, Hanover, NH, United States

1 Introduction

Security problems often stem from one or more mismatches between:

- a person's belief about something, e.g., a clinician may believe that a patient has a weight based on a patient weighing that they conduct;
- how that thing is represented or recorded within, say, a system or a document, e.g., the clinician may record the patient's weight within a hospital system, which is rounded to the nearest whole number; and
- the reality of that thing, e.g., the patient's true weight at the time of recording.

*This chapter revises and extends Chapter 7 of Kothari's doctoral thesis (Kothari, 2020). It also builds upon and borrows text from our previous publications (Anantharaman et al., 2020; Kothari, Blythe, Smith, & Koppel, 2018; Smith, Koppel, Blythe, & Kothari, 2015).

https://doi.org/10.1016/B978-0-323-90570-1.00008-5

Copyright © 2022 Elsevier Inc. All rights reserved.

These sorts of mismatches may be harmless or they may be the seed of a security or privacy issue. Consider some examples:

Security experts' failed assumptions. Security experts often design password composition policies for services. These policies impose requirements on user-generated passwords such as requiring a minimum length or requiring special characters. Often, however, practitioners overlook the induced frustration associated with these requirements and how that frustration may translate to user circumvention that nullifies envisioned security gains.

Mismatches in protocol specification and implementation. *A communication protocol* is a set of procedures that specify how two or more entities should communicate, for example, internet-connected devices use such protocols to communicate with other devices or systems. However, the protocol development process can be complex, involving multiple stages and skills associated with goal planning, specification design, community feedback, implementation, and review. This complexity often means that the *protocol designer* who designs the *protocol specification*—the document that specifies the how the implemented protocol should work—is distinct from the *protocol implementer* who *implements* (or writes code for) the protocol in software or hardware based on their conception of the protocol specification. Often, mismatches between the assumptions made by protocol designers and implementers manifest as security vulnerabilities.

For example, consider the Heartbleed bug (Carvalho, DeMott, Ford, & Wheeler, 2014). The Transport Layer Security (TLS) protocol is a communication protocol used in internet-based communication. The protocol specification supports a special message, called the Heartbeat message; a device ensures that a secure connection remains open by successfully sending a Heartbeat *packet* (data that is organized for processing) to the server and receiving a packet with almost identical content. The implementation of the Heartbleed code for OpenSSL did not check the Heartbeat packet size, leading to a vulnerability where an attacker could send a small packet to the server but expect a much larger packet with private data from the server in return. That is, there was a mismatch between the implementer's mental model and what was written in the specification, which was in turn informed by the designer's mental model; the implementer did not include the checks that were envisioned by the protocol designer and specified in the specification. In addition to flawed mental models and oversights on the part of the implementer that lead to vulnerabilities like Heartbleed, specification-implementation mismatches include ambiguous specifications that lend themselves to different interpretations.

System ambiguity. System developers may have preconceived notions about how a user will use their systems. Consider a *data entry system*, a system that users use to record and often read data. The developer of such a system may overlook the reality that different systems of measurements are used in different countries, resulting in a data field within the data entry system for which no unit of measurement is specified. A user may enter data into the system assuming one system of measurements (e.g., the metric system), which is later read by someone who assumes another system of measurements (e.g., US customary system). Hence, an implicit assumption on behalf of the developer may introduce ambiguity into the system. For one such example, we have heard reports of IT systems in a hospital using a different measurement system from that country's standard—risking serious consequences for patients.

Hidden behavior. Another example involves a user making a search query on Google. The user sees a URL—a sequence of characters that specifies the address of a resource on the internet or a network more generally—in the status bar of their web browser, and they click on the corresponding hyperlink. In reality, the user navigates through another Google URL before (*hopefully* (Abrams, 2021)) ending up on the page they initially intended, none the wiser if all goes well (Cyphers, Miagkov, & Arrieta, 2018). The user is unlikely to notice this redirection, which very well may violate their privacy expectations.

Programmers' tacit assumptions. Programmers often use the phrase *DRY*, which means do not repeat yourself (Oriol Salides, 2021), with the aim of discouraging the reuse of previously written, presumably more reliable, code. Developers make routine software tasks available as software libraries for others to use. These software libraries have an *application programming interface* (API) (Wikipedia Contributors, 2021 a), which is a document that specifies, among other things, how the library should be used. Instead of consulting the API, programmers may assume the function behavior based on prior experience, which may result in unintended behavior being introduced into the resultant program.

Privacy expectations versus reality. Several recent studies have focused on understanding users' privacy expectations pertaining to internet-of-things (IoT) devices, which correspond to the multitude of highly communicative internet-connected devices (Naeini et al., 2017). Various researchers have used the contextual integrity (CI) framework to study these privacy norms (Nissenbaum, 2009). This framework defines privacy as the *appropriate flow* of information subject to contextual information. In this framework, privacy norms change over time and from context to context. Any flow that is not aligned with the well-established legal, ethical, or standard practice norms is considered a *privacy violation*. Based on the CI framework, privacy violations stem from mismatches between users' expectations and device behavior.

For example, Malkin et al. (2019) studied users' beliefs and expectations of their smart speakers (or smart voice assistants, such as Google Home or Amazon Echo). A study involving 116 participants who use smart speakers found that many users considered the music they listened to be private. In contrast, the default setting on many music applications allows others to know what song the user is listening to. In another finding, due to unreliable voice activation, users found unintended recordings of their children or grandchildren. These examples highlight mismatches between device behavior and users' expectations of device behavior. That is, they reveal privacy violations.

In fact, across our work, we find it exceedingly difficult to find instances of security and privacy issues where we *cannot* identify such an underlying disconnect. If we dig deep enough, security problems tend to come down to one or more of these mismatches or, more precisely, what we call *mismorphisms*—"mappings that *fail* to preserve structure" (Smith et al., 2015).

If mismorphisms indeed lie at the heart of security and privacy issues, then understanding mismorphisms, developing a suitable model to express them, and working toward a rich catalog of them may serve useful in addressing the security and privacy issues to which they lead. This chapter reviews our efforts and our colleagues' efforts in achieving these aims.

This chapter is organized as follows. In Section 2, we provide a brief primer on semiotics. In Section 3, we introduce a semiotic model to capture mismorphisms that adopts a variation of semiotic triads. In Sections 4 and 5, we motivate the development of an alternative logic model to capture mismorphisms and we present said model. In Section 6, we use the logic model to develop a preliminary classification of a number of mismorphisms. In Section 7, we discuss future work. In Section 8, we conclude.

2 A brief background on semiotics

Semiotics is the study of signs, processes that involve signs, and how meaning is conveyed through signs (Wikipedia Contributors, 2021 e). A sign may be a sound, an image, a smell, or anything else from which a sentient being extracts meaning. For a simple example, a person may see a stop sign while driving and know that means they should slow down and come to a stop. Semiotic models aim to explain these and other phenomena. Two of the most prevalent semiotic models are: the dyadic model proposed by Ferdinand de Saussure, which includes a signifier and a signified; and the triadic model proposed by Charles Sanders Peirce, which includes a sign, an object, and an interpretant (Chandler, 1994). The

Stanford Encyclopedia of Philosophy (Atkin, 2013) provides a primer on Peirce's work.

Ogden and Richards presented the *semiotic triad* (Ogden & Richards, 1923; Wikipedia Contributors, 2021 g) to capture the relationship between three nodes: the referent (the thing being referred to), the thought or reference (the object evoked by the referent), and the symbol (the object used to represent the thought), as seen in Fig. 1. When a writer writes, the referent—the thing the writer is trying to express—induces a thought based on the writer's knowledge of language, who the writer thinks the reader will be and how they might interpret it, the writer's state of mind, and so forth. The thought evokes a symbol that is supposed to express the referent. Similarly, when a reader reads a word, the word or symbol evokes a thought based on the reader's general knowledge, their understanding of the context in which the word is used, and so forth. The thought is then internalized as a referent. A causal relation is established between the word (the symbol) and the thought (the reference). And a relation is also established between the thought (the reference) and the referent. However, there is no direct relation between the symbol and the referent. Instead, there is an imputed relation established through the two sides of the triangle, not the base. Thus, we have the semiotic triad.

Before discussing our earlier work in building a semiotic triad for mismorphisms, we review some related work at the intersection of semiotics and HCI. Weir (1992) discusses the need for applying semiotic approaches to better understand man-machine communication. de Souza, Barbosa, and Prates (2001) outline desired software design properties, advocate for using semiotic engineering for HCI, and outline one approach.

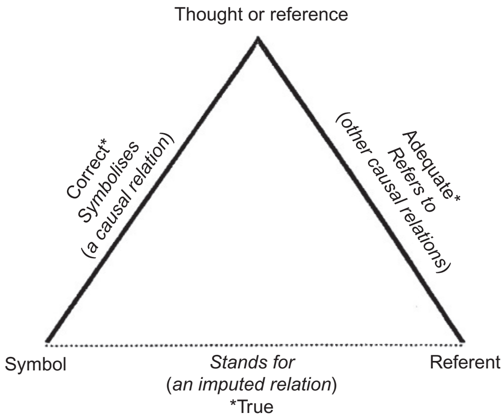

FIG. 1 The semiotic triad. *A slightly modified version of the image appearing on page 11 of the original 1923 publication of Ogden and Richards'* The Meaning of Meaning *(Ogden & Richards, 1923).*

Ferreira, Barr, and Noble (2005) look at how semiotics can be used to understand redesigns of user interfaces. They analyze three instances of redesign of a sign that is part of a user interface, and they briefly look at the contributing factors and propose that such examinations can lead to better user interface design. Andersen (2001) enumerates a number of challenges that semiotics-based HCI design can help address, including: “making HCI more coherent,” “exploiting insights from older media,” “defining the characteristic properties of the computer medium,” and “situating the HCI-systems in a broader context.”

3 A semiotic model for mismorphisms

In this section, we (very) briefly review our earlier work on mismorphisms (Smith et al., 2015). The usage of mismorphism in this section is slightly different from the usage in the logic model we later discuss. However, the essence of the two models are the same; in both models, we seek to capture differences between representations of things that produce security and privacy problems in practice.

As noted in the previous section, our semiotic model of mismorphisms is inspired by Ogden and Richards’ semiotic triad. Our model is built with the intent of expressing circumvention scenarios. We replace the referent with a reality, the thought with a mental model, and the symbol with an IT representation as seen in Fig. 2. In this model,

- the reality corresponds to some truth in the real world, for example, what actions a user may perform;
- the mental model corresponds to a party’s beliefs regarding what the reality should be, for example, what an admin thinks a user’s permissions should be; and
- the IT system representation expresses the reality as expressed in the IT system, for example, what permissions are given to a user.

Unlike Ogden and Richards’ semiotic triad, each side of the triangle now exists and links two nodes. However, this linkage is unidirectional

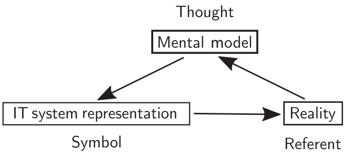

FIG. 2 A triad for capturing circumvention scenarios.

and expresses a single mapping between representations: The reality informs the mental model. A change in the mental model may drive a party to change the IT system itself. And a change to the IT system generates a new reality. For example, a security administrator may observe a reality in which users leave a machine unattended. This observation leads the security practitioner to the belief that there should be automatic timeouts. Thus, the security administrator may implement a policy within the IT system that automatically logs the user out if the IT system detects the user is away. And this, in turn, creates a new reality. Now, the user may become dissatisfied with this reality because the logouts get in the way performing their primary task. This reality will then drive the user to think of a way to circumvent the system. The user may then modify the IT system by, say, attaching a mouse jiggler to the computer, which in turn generates a new reality. (In our original paper (Smith et al., 2015), there was a unidirectional relation from the mental model to the IT system representation. However, in practice, this may be bidirectional. That is, the IT system representation may inform one's mental model. This can be an important source of security vulnerabilities if security personnel rely not on the reality but the IT system representation of the reality to make future decisions.)

In semiotics, the constructs of interest are called *morphisms* and they capture instances where a *predicate*—an expression that can be evaluated as true or false—holds the same truth value across representations. However, as highlighted in the example we just discussed, we are interested in instances where predicates take on different truth values across nodes of the triad. We call these *mismorphisms*. In our initial mismorphism work (Smith et al., 2015), we focused on exploring different classes of mismorphisms and cataloging them using the semiotic model we had developed. We found the model extremely effective in classifying circumvention scenarios.

4 Beyond semiotic triads

At the beginning of this chapter, we said mismorphisms are powerful enough to capture the underlying causes of several security and privacy issues. In the previous section, we presented a semiotic model to represent mismorphisms, one which we had used in prior work (Smith et al., 2015) to develop a catalog of practical unusability and circumvention scenarios, primarily in the hospital setting. In this section, we justify an alternative logic model that takes a more general approach to capture mismorphisms as a high-level concept. The development of this model was driven by the following considerations:

- While we showed that the semiotic approach to modeling mismorphisms is effective for capturing a number of usable security scenarios (Smith et al., 2015), it is somewhat constrained and things can get messy with additional complexity, for example, when
  - there is more than one single human interacting with a single system,
  - there are temporal effects, or
  - multiple mismorphisms are at play.

  This does not mean that we cannot use the notion of semiotic triads to capture complex scenarios. Indeed, we had demonstrated how we actually can capture some of these phenomena using the semiotic model of mismorphisms (Smith et al., 2015). However, the model does not naturally lend itself to such complexity. Ideally, we want to construct a model that has the machinery to both succinctly and accurately capture such mismorphism complexity.
- In our semiotic model, we used mathematical logic to express the underlying predicates. Reenvisioning the notion of mismorphisms using a pure logic model while retaining much of the spirit of the initial semiotic model seems natural.
- At its core, a mismorphism constitutes a difference in the interpretation of something by two or more entities. The model should allow us to capture mismorphisms involving any integer-valued number of interpretations above 1.
- This model should lay the foundation for a mismorphism submission interface that supports collaborative community development of a classification framework with accompanying examples; ergo, ease of representation of mismorphisms in the given model is paramount.

These considerations motivated us to develop a logic model for mismorphisms that is built on the notion of interrepresentational differences, that is intuitive, and that supports multiple representations, temporality, and compositionality.

5 A logic for mismorphisms[a]

The limitations discussed in the previous section drove us to develop a new model of mismorphisms, one grounded not in semiotic triads but in logic. In this section, we present this logic model. In the next section, we demonstrate how we can use this logic to create a catalog of the underlying causes of security and privacy issues.

[a] This section revises and extends text from our previous paper (Anantharaman et al., 2020).

For the rest of this chapter, we will refer to a *mismorphism* as a difference in interpretation of a predicate between two or more interpreters. That is, we can think of different *interpreters* (e.g., a person, a system, a document, code) interpreting propositions or predicates about the world. In general, it is good when the interpretations agree and are in accordance with reality. However, when a predicate takes different truth values across different interpretations, we have a mismorphism. Not all mismorphisms are bad; some may be benign or even beneficial. However, many times, mismorphisms produce unintended or undesirable outcomes that undermine security and privacy objectives at both the individual and organization level. This high-level vision for what constitutes a mismorphism drives the development of the formal logic we use to capture them.

Within our logic, we use the words *predicate* and *interpretation* in similar—albeit, not identical—manners to the common formal-logic meanings, for example, as presented by Aho and Ullman (1994). However, instead of a binary logic, we use a ternary logic similar to Kleene's ternary logic (Fitting, 1994; Goodstein, 1954).[b] We refer to a *predicate* as a function of zero or more variables whose codomain is $\{T, F, U\}$ where T is true, F is false, and U is uncertain/unknown. We refer to an *interpretation* of a predicate as an assignment of values (which may include U) to variables, which results in the predicate being interpreted as T, F, or U. A predicate is interpreted as T if after substituting all variables for their truth values, the predicate is determined to be T; it is interpreted as F if after substituting all variables for their truth values, the predicate is determined to be F; if we are unable to determine whether the predicate is T or F by substitution, the predicate is interpreted as U to signify that the truth value of the predicate cannot be determined.

The interpretation must be done by someone (or perhaps something, e.g., a system or a specification) and that entity is called the *interpreter*. In our model, for instances where there is some ground truth, we have a special interpreter, the oracle O, who interprets the predicate as it is in reality. Some interpreters may lack the requisite information to assign values to variables that would result in the predicate being interpreted as T or F. In these instances the predicate should be interpreted as U. We use $P|_A$ to denote the interpretation of predicate P by interpreter A.

To represent mismorphisms, we need a way to express scenarios where two or more interpreters diverge in their interpretations of a predicate. That is we must define relations on the interpreters' interpretations of a predicate. Ergo, we introduce the notions of *interpretation relations* and

[b] We do not specify a specific ternary logic system for evaluating predicates in this chapter.

interpretative expressions. In the context to our model, a *simple interpretative expression* has the form:

Predicate (Interpretation Relation) Interpreters

Here, the *interpretation relation* is a k-ary relation where $k \geq 2$ denotes the number of interpreters involved—and the k-ary relation is over the interpretations of the predicate by the k interpreters. The three classes of interpretation relations that we are concerned with in this chapter are: the interpretation-equivalence relations $\left(\boxed{\underset{\text{interp}}{=}}\right)$, the interpretation-uncertainty relations $\left(\boxed{\overset{?}{\underset{\text{interp}}{=}}}\right)$, and the interpretation-inequivalence relations $\left(\boxed{\overset{\times}{\underset{\text{interp}}{=}}}\right)$.[c] The interpretation relations[d] we examine are defined as follows, where each P represents a predicate and each A_i represents an interpreter:

- $P\left(\boxed{\underset{\text{interp}}{=}}\right), A_1, A_2, \ldots, A_k$ if and only if P, as interpreted by each A_i, has a truth value that's either T or F (never U)—and all interpretations yield the same truth value. The interpretation-equivalence relations correspond to situations where every interpreter can evaluate the predicate as T or F and they all evaluate it in the same way.

- $P\left(\boxed{\overset{?}{\underset{\text{interp}}{=}}}\right) A_1, A_2, \ldots, A_k$ if and only if P takes on the value U when interpreted by at least one A_i. The interpretation-uncertainty relations correspond to situations where at least one interpreter lacks the requisite information to fully resolve the predicate to a known truth value of T or F.

- $P\left(\boxed{\overset{\times}{\underset{\text{interp}}{=}}}\right) A_1, A_2, \ldots, A_k$ if and only if P interpreted by A_i is T and P interpreted by A_j is F for some $i \neq j$. The interpretation-inequivalence relations correspond to situations where it can be determined that two interpreters disagree on the interpretation of a predicate.

There are a few important observations to note. One is that the oracle O always holds the correct truth value for the predicate by definition.

[c] Note that for $k=2$, if we confine ourselves to predicates that only take on T or F values, the relation $\underset{\text{interp.}}{=}$ is an equivalence relation in the mathematical sense, as one might expect, that is, it obeys reflexivity, commutativity, and transitivity.

[d] Technically, these are classes of interpretation relations, but adopting this terminology would be prohibitively tedious for us to write that and for you to read.

Another is that if we only know the $\left(\boxed{\underset{\text{interp}}{\overset{?}{=}}}\right)$ relation applies, we will not know which interpreter is uncertain about the predicate or even how many interpreters are uncertain unless $k = 2$ and one interpreter is the oracle. Similarly, if we only know that the $\left(\boxed{\underset{\text{interp}}{\overset{\times}{=}}}\right)$ relation applies, we do not know where the mismatch lies unless $k = 2$. That said, knowledge that the oracle O always holds the correct interpretation, where we are dealing with facts, combined with other information can help specify where the uncertainty or inequivalence stems from. Of course, the formalism could also be changed to allow a bit more flexibility here, but we did not see the need. Last, the $\left(\boxed{\underset{\text{interp}}{=}}\right)$, relation will not be true if the $\left(\boxed{\underset{\text{interp}}{\overset{?}{=}}}\right)$ or the $\left(\boxed{\underset{\text{interp}}{\overset{\times}{=}}}\right)$ relations are true; however, $P \left(\boxed{\underset{\text{interp}}{\overset{?}{=}}}\right) A_1, \ldots, A_k$ and $P \left(\boxed{\underset{\text{interp}}{\overset{\times}{=}}}\right) A_1, \ldots, A_k$ may simultaneously be true.

In addition to the simple interpretative expressions mentioned previously, there are instances where it is useful to consider the composition of simple interpretative expressions. We call such expressions *compound interpretative expressions* and they are expressed by linking together simple interpretative expressions with any number of *interpretative operators* including:

- $\wedge$: The *and* operator has two operands, each of which are either simple or compound interpretative expressions; the full expression applies when both operands apply.
- $\vee$: The *or* operator has two operands, each of which are either simple or compound interpretative expressions; the full expression applies when at least one of the two operands apply.

Additional interpretative operators may serve valuable in creating a larger corpus, but for the purpose of our preliminary catalog of mismorphisms, these will suffice.

Let us revisit the initial goal in developing this logic model: to capture mismorphisms. Within this logic model, a *simple morphism* is a simple interpretative expression where the interpretation-equivalence relation applies. A *simple mismorphism* is a simple interpretative expression where the interpretation-equivalence relation does not apply; equivalently, it is a simple interpretative expression in which the interpretation-uncertainty relation applies or the interpretation-inequivalence relation applies. More generally, a *mismorphism* is either a simple mismorphism or a *compound interpretative expression* for which *at least* one simple mismorphism *must* hold for the expression to hold.

There are some natural extensions to this logical formalism. In select cases, we may want to consider multiple interpreters of the same role. In these instances, we could assign subscripts to distinguish roles, for example, D, I_1, I_2, O might correspond to the developer of a protocol, two implementors of the protocol, and the oracle. Also, there are temporal aspects that may be relevant. Predicates can be functions of time and so can the interpretations. While we use the v^t-style notation to represent a variable as a function of time within a predicate, we may also consider the interpreter as a function of time, for example, $I_4^{t_3}$ means the interpretation is done by implementor I_4 at time t_3. We do not use all of these extensions in this presentation, but if we were to create a larger catalog, they would serve useful.

6 A preliminary catalog of mismorphisms[e]

In this section, we discuss numerous examples of mismorphisms, classified by their general form. First, some remarks:

- The categories are not disjoint. Some mismorphisms could be placed within two or more categories.
- Some mismorphisms may be linked. For example, one mismorphism may lie at the heart of another or perhaps two mismorphisms contribute to a single security issue. This makes sense as many security issues have multiple layers of complexity. We discuss this issue more in the following section.
- We also note that there are multiple ways to do this classification. For example, another natural approach may be to choose the categories based on their application domain or security and privacy context.
- Our focus here is on applying our mismorphisms logic to a diverse but small set of security and privacy issues in different domains. Other flavors of mismorphisms certainly exist as well.

6.1 Breakdown of implication

In certain circumstances, an interpreter may believe a conditional statement that fails to hold in practice—or vice versa, they may not believe a conditional statement holds when it does hold in practice. That is, we may have something of form:

$$(X \implies Y) \quad \boxed{\underset{\texttt{interp}}{\overset{\times}{\equiv}}} \quad A, O$$

[e] This section revises and extends ideas and text from our previous work (Anantharaman et al., 2020; Kothari, 2020; Kothari et al., 2018).

Consider the following examples of this flavor of mismorphism at play:

- A prevailing belief is that users are privacy pragmatists who willingly make informed decisions to give up their privacy in exchange for services (Draper, 2017). This argument, in other words, assumes that the decision to use a service implies the user is making an informed choice. Work by Draper (2017), as well as by others (boyd & Hargittai, 2010; Kokolakis, 2017; Olmstead & Smith, 2017; Turow, Hennessy, & Draper, 2015; Urban & Hoofnagle, 2014) call this view into question. Draper argues that many users feel their privacy is gone and so they resign to giving up control over their data privacy.
- Turow et al. (2015) found that 65% of respondents to a survey believed that the existence of a privacy policy on a site meant the site would not share their information unless they gave explicit permission.
- It is often assumed that adding a privacy option to a service will only improve users' privacy. However, the user's determination of a privacy option may, itself, leak information. For example, Lewis, Kaufman, and Christakis (2008) note that both the options of sharing information or not sharing information correlate with other demographic information.

Alternatively, the implication operation may be correct, but X may not hold true, meaning nothing can be inferred about Y. (Or perhaps, we may observe the opposite direction where both the relation and X hold in practice but not within someone's mental model.)

$$((X \implies Y) \boxed{\underset{\text{interp}}{=}} A, O) \wedge (X \boxed{\underset{\text{interp}}{\overset{\times}{=}}} A, O)$$

For example,

- A security practitioner may assume that any user of a service who wishes to change their privacy settings will be able to do so if they know about them, and, moreover, they may assume the user is aware of those settings. In some cases, even if the former holds, the latter does not.

6.2 Temporal effects

Time may influence how predicates are evaluated. An individual may lack the foresight to identify these temporal effects.

$$(X^t = X^{t+\delta}) \boxed{\underset{\text{interp}}{\overset{\times}{=}}} A, O$$

Some examples:

- As an employee changes roles, their permissions may accumulate, whereas a security practitioner might expect the permissions to be adjusted according to the role (Sinclair, 2013).

- Time-of-check-time-of-use (toctou) bugs (Wikipedia Contributors, 2021 f) occur when there is a delay between when something is checked and when it is used. The delay means that operations may be performed on input that previously satisfied certain properties, but no longer do so. It reflects an oversight on the part of the developer.
- Shotgun parsing (Bratus, Patterson, & Hirsch, 2013) involves scattering parser code—the code responsible for vetting the input to a program—across a program, which results in code being executed before it is recognized. Vulnerabilities that exist in code that are attributable to the shotgun parser antipattern can be classified under this class of mismorphisms.
- An analog of toctou for the privacy domain is time-of-configure-time-of-use: The user may configure their privacy settings on a social networking service once, when they begin using a service. However, over time, people may join or leave the service, leaving their privacy choices outdated. Available privacy options may also change over time.
- Gaw and Felten argue that the user may choose to reuse a password for an account—i.e., select a weak password before that account is associated with sensitive information—and, by the time that account has accrued information, "they're locked into their reused password" (Gaw & Felten, 2006).
- A similar phenomenon may be true with privacy settings. Namely, the user may choose privacy settings before sensitive information is tied to their account. By the time sensitive information is tied to their account, the user may no longer think about privacy. Moreover, in instances where the user does contemplate reconfiguring privacy settings, there is a possibility that the data in question may be perceived to already be lost and, therefore, not worth protecting.
- On the other hand, some users may have already invested significant time or effort in selecting a piece of software, downloading it, and installing it before they configure their privacy settings, compelling them to continue using the service even if it does not meet their privacy needs. That is, they may fall victim to the sunk cost fallacy (Arkes & Blumer, 1985). Had they known of the invasive privacy settings beforehand, they may have chosen to go with a competitor.

6.3 A knowledge gap

In certain circumstances, an interpreter's lack of knowledge about how to interpret information may contribute to a security issue. Here, *P* may be a statement about, say, a system, and the interpreter may be ill-equipped to evaluate the truth of that statement, resulting in an unknown truth value under their interpretation:

$$P \ \boxed{\underset{\text{interp}}{\overset{?}{=}}}\ A, O$$

- Users may lack the requisite information to make informed decisions, often because that information is simply not available. It is not always clear how services safeguard user data, nor the intricacies of how that data is used in practice. Privacy policies exist but they may be exorbitantly time-consuming to read and difficult to digest (McDonald & Cranor, 2008; Obar & Oeldorf-Hirsch, 2016). Moreover, they are often vague and usually subject to change. A pessimist might argue that in practice many existing interfaces and privacy policies ensure users remain uninformed while presenting the veneer of informed consent, thereby persuading their users and others that user data is in good hands. Another concern is that primary services or third-party services may violate privacy policies, terms of service, or users' privacy expectations. This may even be compounded by a delay in reporting violations. Collectively, these and other factors support the argument that most users do not—and, at least in the current privacy landscape cannot—have a concrete understanding of how their data is used.
- A user may lack the capability to come to a determination regarding the safety of a URL (e.g., shortened URLs, gatekeeper URLs), the legitimacy of an email, or the meaning of certificate information. While some users may seek information that informs their mental model, others may fall back on insecure behavior because it is less effort and potentially a lower perceived cost than alternatives. Even if a user seeks out information, it is possible that they may consult a resource that provides inaccurate information.

6.4 Projections

An interpreter A's interpretation of how interpreter B would interpret a predicate P may differ from how B actually interprets it. That is, we may have

$$P|_B \ \boxed{\underset{\text{interp}}{\overset{\times}{=}}}\ A, O$$

We recognize that there is a slight abuse of notation here. To resolve this, we can simply substitute $P|_B$ with "B's interpretation of P," to avoid the double-meaning of $P|_B$—or we could create a wrapper. Recall the oracle is always correct and so their interpretation of $P|_B$ aligns with what $P|_B$ actually is. In any case, here is an example of such a mismorphism:

- Actual and perceived time and effort to configure privacy settings may influence whether the user begins configuring them and whether they finish. For example, the user may be dissuaded from using an interface that appears illogical, complex, or hard to navigate. Or, as we mentioned earlier, they may simply lack the requisite knowledge to make meaningful decisions that align with their intentions. The security practitioner or others may perceive users' effort to configure their privacy settings to be minimal or ignore them altogether and view the option of configuration as a binary choice.

7 Future work

In this section, we briefly discuss future work pertaining to the logic model of mismorphisms.

7.1 Peeling back the layers of mismorphisms

In the previous section, we presented a preliminary catalog of mismorphisms that captures a number of security and privacy issues across domains. However, much of the power of mismorphisms as an explanatory model comes from the ability to break down a mismorphisms and study their ramifications. Identifying these causal relations allows us deconstruct and learn from existing security problems.

For example, why might a user wrongly classify an unsafe URL as safe? Well, one reason may be that their mental model of where the URL goes is flawed (Albakry, Vaniea, & Wolters, 2020). This can be captured as a mismorphism between security properties of the URL in the system representation and the security properties of the URL within the user's mental representation. But why does that mismorphism exist? It may be a purely visual problem, due to a poor choice of font, which can be expressed as a mismorphism between the user's mental representation and the information shown in the real-world and/or a mismorphism between the system representation and the information shown in the real-world, depending on where the problem lies. Or perhaps the user correctly interprets what characters are on the screen, but fails to extract the correct security information from those characters; this again can be represented as a mismorphism between the URL specification (or, more precisely, the layers of systems involved in resolving the URL and delivering content to the user) and the user's mental representation. But we could again ask: why is there a mismorphism between a user's mental model of URL structure and the way users are resolved in practice? And so on.

Ultimately, it is this process of recursive deconstruction of mismorphisms that reveals why a security problem truly exists. Understanding mismorphisms and the links between them is essential to addressing many of the security and privacy challenges of today. Our logic model already supports some interpretative operators. However, support for casual relations would provide tremendous expressive power. Indeed, the addition of causal relations reintroduces some of the lost expressiveness of semiotic triads without sacrificing any of the expressiveness of the logic model. Approaches such as supplementing the notion of mismorphisms discussed here with, say, events, may provide the best of both models. Support for both, expressing causality and expressing events, are beyond the scope of this chapter, though they are certainly worth pursuing. Together, they would enable us to capture the development of security and privacy issues as a chain of mismorphisms and events linked by causal relations.

7.2 Psychological phenomena and the genesis of mismorphisms

There is an intimate link between psychological phenomena, especially cognitive biases, and mismorphisms. In this chapter, we have treated mismorphisms as the most elementary phenomenon driving security and privacy issues. But this raises the question of why mismorphisms arise. Part of the explanatory process to tackle this question may involve deconstructing mismorphisms into predecessor mismorphisms, as we discussed in the previous subsection. But it may also serve useful to examine the link between human psychology and mismorphisms.

Several researchers have explored the psychological aspects of computer security (Enrici, Ancilli, & Lioy, 2010): Lafrance (2004) categorized hackers based on their motivations and techniques. In comparison, Enrici et al. (2010) explore how humans can be the target of attacks—cognitive hacks. Smith (2012), an author on this chapter, has examined various cognitive biases and how they can be leveraged to improve security.

Yet there are still many avenues at the intersection of psychology and security that researchers have not yet or have only partially explored. Consider cognitive biases, the logical flaws in judgment that humans tend to exhibit. A deep exploration of programmers' cognitive biases could help to explain the underlying causes of programming mistakes that ultimately manifest as security vulnerabilities. For some examples, we believe exploration of three egocentric biases in programming may help to reduce security vulnerabilities in practice. A person who is subject to the overconfidence bias (Wikipedia Contributors, 2021 d) or illusion of validity bias (Wikipedia Contributors, 2021 b) may overestimate their performance on a task. A programmer who is subject to these biases may overestimate the correctness of their beliefs as they write code or the code itself. Another relevant bias here is the illusion of optimism bias, wherein a person believes they will not experience a negative event (Wikipedia Contributors, 2021 c). When a programmer writes code, they must account for infinite inputs, and they should be certain that no input can crash the program or enable unintended computation (Bratus, Locasto, Patterson, Sassaman, & Shubina, 2011). However, the illusion of optimism bias may drive programmers to believe that their code would not be a target of attack, leading them to overlook these checks.

Moreover, to the best of our knowledge, there is a missing explanatory framework linking these psychological phenomena to the security and privacy issues they ultimately give rise to. Augmenting the mismorphisms framework to express the link between these psychological phenomena and mismorphisms would allow us to bridge this gap and produce a holistic explanation of the genesis of security and privacy issues.

7.3 A mismorphism submission interface

The primary aim of the logic model for mismorphisms is to provide a simple, intuitive, and precise framework to accurately catalog how real-world security and privacy issues came to be. We seek to capture the initial starter mismorphisms and the chain of causal linkages that ultimately produce the problem scenarios of interest. A framework that achieves this aim, alongside a catalog with a navigable user interface, can be of great benefit to the security and privacy communities.

7.3.1 Informing design decisions

First, it would inform new design decisions. Practitioners would be able to learn from the mistakes of the past during the design and development of systems, policies, protocols, and so forth. When faced with a design question, having at your disposable a navigable catalog of bad outcomes—and also potentially good outcomes—associated with related design decisions that others have made would be of immense value.

7.3.2 Addressing existing security and privacy problems

Second, having such a framework can help in conceptualizing and addressing existing security and privacy problems faced by organizations and individuals. Moreover, if we augment the catalog by allowing users to specify how they attempted to address given mismorphisms, as well as comment on how effective those methods were, this can provide additional insights to those facing similar mismorphisms.

7.3.3 Informing best practices

As mismorphisms are often tethered to the human and these links can be readily observed upon deconstructing mismorphisms, a catalog of mismorphisms can shed light on human failings. Best practices can then be developed to accommodate these failings. Given the sheer breadth of security and privacy issues, however, cataloging mismorphisms would be a mammoth undertaking that would require a community-driven effort. The most natural way to achieve this would be to develop an online mismorphism submission interface.

We envision an online interface that allows users to

- submit mismorphism classes by specifying
  - a mismorphism class title;
  - a representation of the mismorphism class using logic;
  - relations between the target mismorphism class and other mismorphism classes (e.g., member of, produces);
  - a verbal description of the mismorphism class; and
  - auxiliary data (e.g., references).
- submit examples of security and privacy issues that stem, at least partially, from a mismorphism belonging to a user-specified mismorphism class;
- submit information regarding solutions adopted to tackle mismorphisms and the security and privacy issues they give rise to, as well as information on the efficacy of such solutions; and
- view information about a class of mismorphisms, including the aforementioned data submitted by the user, a visual representation of the mismorphism class, links to other mismorphism classes, and example security and privacy issues tied to the mismorphism class.

8 Conclusion

In this chapter, we pursued a logic model of mismorphisms to complement our earlier work on capturing mismorphisms via semiotic triads. We reviewed the earlier semiotic triad model, provided our rationale for developing a new model, introduced our logic model, cataloged a variety of mismorphisms, and discussed future work.

References

Abrams, L. (2021). *Beware: Malicious home depot ad gets top spot in Google search*. Bleeping Computer. https://www.bleepingcomputer.com/news/security/beware-malicious-home-depot-ad-gets-top-spot-in-google-search/.

Aho, A., & Ullman, J. (1994). *Predicate logic*. Foundations of Computer Science. http://infolab.stanford.edu/~ullman/focs.html.

Albakry, S., Vaniea, K., & Wolters, M. K. (2020). What is this URL's destination? Empirical evaluation of users' URL reading. In *Proceedings of the 2020 CHI conference on human factors in computing systems* (pp. 1–12). New York, NY: Association for Computing Machinery. https://doi.org/10.1145/3313831.3376168.

Anantharaman, P., Kothari, V., Brady, J. P., Jenkins, I. R., Ali, S., Millian, M. C., & Smith, S. W. (2020). Mismorphism: The heart of the weird machine. In *Security protocols XXVII* (pp.

113–124). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-57043-9_11.

Andersen, P. B. (2001). What semiotics can and cannot do for HCI. *Knowledge-Based Systems*, *14*(8), 419–424. https://doi.org/10.1016/S0950-7051(01)00134-4.

Arkes, H. R., & Blumer, C. (1985). The psychology of sunk cost. *Organizational Behavior and Human Decision Processes*, *35*(1), 124–140. https://doi.org/10.1016/0749-5978(85)90049-4.

Atkin, A. (2013). Peirce's theory of signs. In E. N. Zalta (Ed.), *The Stanford encyclopedia of philosophy* Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/sum2013/entries/peirce-semiotics/ (Ed.).

Boyd, D., & Hargittai, E. (2010). Facebook privacy settings: Who cares? *First Monday*, *15*(8). https://doi.org/10.5210/fm.v15i8.3086.

Bratus, S., Locasto, M., Patterson, M., Sassaman, L., & Shubina, A. (2011). Exploit programming: From buffer overflows to "weird machines" and theory of computation. *USENIX; login:*, *36*(6), 13–21.

Bratus, S., Patterson, M. L., & Hirsch, D. (2013). From "shotgun parsers" to more secure stacks. *Shmoocon*. https://www.cs.dartmouth.edu/~sergey/langsec/ShotgunParsersShmoo.pdf.

Carvalho, M., DeMott, J., Ford, R., & Wheeler, D. A. (2014). Heartbleed 101. *IEEE Security & Privacy*, *12*(4), 63–67. https://doi.org/10.1109/MSP.2014.66.

Chandler, D. (1994). *Signs*. Semiotics for beginners. http://visual-memory.co.uk/daniel/Documents/S4B/sem02.html.

Cyphers, B., Miagkov, A., & Arrieta, A. (2018). *Privacy badger now fights more sneaky Google tracking*. https://www.eff.org/deeplinks/2018/10/privacy-badger-now-fights-more-sneaky-google-tracking.

de Souza, C. S., Barbosa, S. D. J., & Prates, R. O. (2001). A semiotic engineering approach to user interface design. *Knowledge-Based Systems*, *14*(8), 461–465. https://doi.org/10.1016/S0950-7051(01)00136-8.

Draper, N. A. (2017). From privacy pragmatist to privacy resigned: Challenging narratives of rational choice in digital privacy debates. *Policy & Internet*, *9*(2), 232–251. https://doi.org/10.1002/poi3.142.

Enrici, I., Ancilli, M., & Lioy, A. (2010). A psychological approach to information technology security. In *3rd international conference on human system interaction* (pp. 459–466). https://doi.org/10.1109/HSI.2010.5514528.

Ferreira, J., Barr, P., & Noble, J. (2005). The semiotics of user interface redesign. In *Proceedings of the sixth Australasian conference on user interface: Vol. 40* (pp. 47–53).

Fitting, M. (1994). Kleene's three valued logics and their children. *Fundamental Information*, *20*(1–3), 113–131. http://dl.acm.org/citation.cfm?id=183529.183533.

Gaw, S., & Felten, E. W. (2006). Password management strategies for online accounts. In *Proceedings of the second symposium on usable privacy and security* (pp. 44–55). https://doi.org/10.1145/1143120.1143127.

Goodstein, R. (1954). Introduction to Metamathematics. By S. C. Kleene. Pp. x, 550, Fl. 32.50. 1952. (Noordhoff, Groningen; North-Holland Publishing Co., Amsterdam). *The Mathematical Gazette*, *38*(323), 72–76. https://doi.org/10.2307/3609805.

Kokolakis, S. (2017). Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon. *Computers & Security*, *64*, 122–134. https://doi.org/10.1016/j.cose.2015.07.002.

Kothari, V. (2020). Bridging the gap between intent and outcome: Knowledge, tools & principles for security-minded decision-making. Dartmouth College, PhD Dissertation. *Dartmouth Digital Commons*, *60*. https://digitalcommons.dartmouth.edu/dissertations/60.

Kothari, V., Blythe, J., Smith, S., & Koppel, R. (2018). Data privacy and the elusive goal of empowering the user. In *Workshop on moving beyond a 'one-size-fits-all' approach: Exploring individual differences in privacy, CHI*.

Lafrance, Y. (2004). Psychology: A precious security tool. *SANS GSEC Certification, Practical Assignment*. https://www.sans.org/reading-room/whitepapers/engineering/psychology-a-precious-security-tool-1409.

Lewis, K., Kaufman, J., & Christakis, N. (2008). The taste for privacy: An analysis of college student privacy settings in an online social network. *Journal of Computer-Mediated Communication*, *14*(1), 79–100. https://doi.org/10.1111/j.1083-6101.2008.01432.x.

Malkin, N., Deatrick, J., Tong, A., Wijesekera, P., Egelman, S., & Wagner, D. (2019). Privacy attitudes of smart speaker users. *Proceedings on Privacy Enhancing Technologies*, *2019*(4), 250–271. https://doi.org/10.2478/popets-2019-0068.

McDonald, A. M., & Cranor, L. F. (2008). The cost of reading privacy policies. *ISJLP*, *4*, 543.

Naeini, P. E., Bhagavatula, S., Habib, H., Degeling, M., Bauer, L., Cranor, L. F., & Sadeh, N. (2017). Privacy expectations and preferences in an IoT world. In *Thirteenth symposium on usable privacy and security (SOUPS 2017)* (pp. 399–412). Santa Clara, CA: USENIX Association.

Nissenbaum, H. (2009). *Privacy in context: Technology, policy, and the integrity of social life*. Stanford University Press. https://doi.org/10.5860/choice.47-6940.

Obar, J. A., & Oeldorf-Hirsch, A. (2016). The biggest lie on the Internet: Ignoring the privacy policies and terms of service policies of social networking services (June 1, 2018). *Information, Communication & Society* (pp. 1–20). 2018. TPRC 44: The 44th Research Conference on Communication, Information and Internet Policy. https://doi.org/10.2139/ssrn.2757465.

Ogden, C. K., & Richards, I. A. (1923). *The meaning of meaning: A study of the influence of language upon thought and of the science of symbolism*. London, K. Paul, Trench, Trubner & Co.; New York, Harcourt, Brace & Co. https://doi.org/10.1038/111566b0.

Olmstead, K., & Smith, A. (2017). *Americans and cybersecurity*. Pew Research Center. https://www.pewresearch.org/internet/2017/01/26/americans-and-cybersecurity/.

Salides, Oriol. (2021). DRY: Don't repeat yourself. https://apiumhub.com/tech-blog-barcelona/dry-dont-repeat-yourself/. Accessed 31 March 2021.

Smith, S. W. (2012). Security and cognitive bias: Exploring the role of the mind. *IEEE Security & Privacy*, *10*(5), 75–78. https://doi.org/10.1109/MSP.2012.126.

Smith, S. W., Koppel, R., Blythe, J., & Kothari, V. (2015). Mismorphism: A semiotic model of computer security circumvention (extended version). *Computer Science Technical Report TR2015-768*. https://digitalcommons.dartmouth.edu/cs_tr/368.

Sinclair, S. (2013). Access control in and for the real world. Dartmouth College, PhD Dissertation. *Dartmouth Digital Commons*, *43*. https://digitalcommons.dartmouth.edu/dissertations/43.

Turow, J., Hennessy, M., & Draper, N. A. (2015). The tradeoff fallacy: How marketers are misrepresenting American consumers and opening them up to exploitation. *Available at SSRN 2820060*. https://doi.org/10.2139/ssrn.2820060.

Urban, J., & Hoofnagle, C. (2014). The privacy pragmatic as privacy vulnerable. In *Symposium on usable privacy and security (SOUPS 2014)*. Berkeley, CA: USENIX Association. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2514381.

Weir, G. R. S. (1992). Meaningful interaction in complex man-machine systems. *Reliability Engineering & System Safety*, *38*(1–2), 151–156. https://doi.org/10.1016/0951-8320(92)90116-3.

Wikipedia, Contributors. (2021a). *API—Wikipedia, the free encyclopedia*. https://en.wikipedia.org/w/index.php?title=API&oldid=1014458682. Accessed 18 April 2021.

Wikipedia, Contributors. (2021b). *Illusion of validity—Wikipedia, the free encyclopedia*. https://en.wikipedia.org/w/index.php?title=Illusion_of_validity&oldid=999566863. Accessed 18 April 2021.

Wikipedia, Contributors. (2021c). *Optimism bias—Wikipedia, the free encyclopedia*. https://en.wikipedia.org/w/index.php?title=Optimism_bias&oldid=1011001205. Accessed 18 April 2021.

Wikipedia, Contributors. (2021d). *Overconfidence effect—Wikipedia, the free encyclopedia*. https://en.wikipedia.org/w/index.php?title=Overconfidence_effect&oldid=1011068721. Accessed 18 April 2021.

Wikipedia, Contributors. (2021e). *Semiotics—Wikipedia, the free encyclopedia*. https://en.wikipedia.org/w/index.php?title=Semiotics&oldid=1017297510. Accessed 18 April 2021.

Wikipedia, Contributors. (2021f). *Time-of-check to time-of-use—Wikipedia, the free encyclopedia*. https://en.wikipedia.org/w/index.php?title=Time-of-check_to_time-of-use&oldid=1014091278. Accessed 18 April 2021.

Wikipedia, Contributors. (2021g). *Triangle of reference—Wikipedia, the free encyclopedia*. https://en.wikipedia.org/w/index.php?title=Triangle_of_reference&oldid=1017387319. Accessed 18 April 2021.