2015 IEEE 28th Computer Security Foundations Symposium

A Methodology for Information Flow Experiments

Michael Carl Tschantz
International Computer Science Institute
mct@icsi.berkeley.edu

Amit Datta
Carnegie Mellon University
amitdatta@cmu.edu

Anupam Datta
Carnegie Mellon University
danupam@cmu.edu

Jeannette M. Wing
Microsoft Research
wing@microsoft.com

***Abstract*—Information flow analysis has largely focused on methods that require access to the program in question or total control over an analyzed system. We consider the case where the analyst has neither control over nor a white-box model of the analyzed system. We formalize such limited information flow analyses and study an instance of it: detecting the usage of data by websites. We reduce these problems to ones of causal inference by proving a connection between noninterference and causation. Leveraging this connection, we provide a systematic black-box methodology based on experimental science and statistical analysis. Our systematic study leads to practical advice for detecting web data usage, a previously unformalized area. We illustrate these concepts with a series of experiments collecting data on the use of information by websites.**

***Keywords*-information flow analysis; causation; online tracking; blackbox experiments;**

I. Introduction

Web Data Usage Detection: Suppose you are shown a car ad by Google while reading an article on a news webpage. You might wonder whether the ad appears because you visited a car dealer website earlier in the day. That is, you would like to know whether information flows from the car dealers website (to the ad network) to the news webpage.

More generally, concerns about privacy (e.g., [1]) have led to much interest in determining whether ad networks, such as Google's DoubleClick and the Yahoo Ad Exchange, use certain types of information [2]–[9]. We call this problem *web data usage detection* (WDUD). In this paper, we show how to conduct experiments in a systematic way that can help answer this and other kinds of privacy-related questions.

While WDUD studies are, in essence, attempting to track the flow of information from inputs to some system to outputs of it, they differ from traditional information flow analyses (IFAs). The traditional motivation for IFA, designing secure programs, leads to viewing the analyst as verifying that a system under his control protects information sensitive to the operator of the system. Thus, the problems studied and analyses proposed tend to presume that the analyst has access to the program running the system in question or total control over its inputs and environment. (See [10] for a survey.)

In the setting of WDUD, the analyzed system can be adversarial toward the person studying the system. The analyst may be aligned with (or even equal to) a *data subject*, an entity whose information is collected by the system. In this setting, the analyst has no access to the program running the system in question, little control over its inputs, and a limited view of its behavior. Thus, the analyst lacks the abilities presupposed by traditional IFAs. To understand the WDUD problem as an instance of IFA requires a fresh perspective on IFA.

The original motivation underlying much of IFA research also obscures its connection to other areas of research. For example, copyright protection [11], [12], traitor tracing [13], data leak detection [14]–[18], and the detection of plagiarism [19] are all in essence information flow analyses in which the analyst has limited access to the system in question (often a person). However, to keep the presentation clear, we focus on WDUD, leaving a discussion of related areas to future work.

Contributions: We develop a formal methodology for conducting IFA for black-box information flow problems, and for WDUD in particular. The overarching contribution of this work is relating IFA in these nontraditional settings to experiments designed to determine causation. We show that the ability of the analyst to control some inputs enables *information flow experiments* that manipulate the system in question to discover the use of information without a white-box model of the system. We present an easy-to-apply, statistically rigorous methodology for information flow experiments that future studies on WDUD and other IFAs for black boxes may use to draw statistically sound conclusions.

Our methodology is supported by a chain of contributions that follows the paper's outline:

- § II a systematization of black-box IFA problems
- § III a proof of a connection between IFA and causality
- § IV an experimental design leveraging this connection
- § V a rigorous statistical analysis for such experiments

In particular, while the link has been subject to prior comment [20], [21], we believe we are the first to formally prove a connection between standard notions of information flow and causation. We are also the first to provide a method for conducting WDUD studies that comes with a methodology showing that a positive result entails a flow of information with a quantified certainty.

These contributions are each necessary for creating a chain of sound reasoning from intuition about vague prob-

DOI 10.1109/CSF.2015.40

lems to rigorous quantified results. This chain of reasoning provides a systematic, unifying view of these problems, which leads to a concrete methodology based on well studied scientific methods. While the notion of experimental science is hardly new, our careful justification provides guidance on the choices involved in actually conducting an information flow experiment.

The systematization of experimental approaches to security, privacy, and accountability is becoming increasingly important as technological trends (e.g., Cloud and Web services) result in analysts and auditors having limited access to and control over systems whose properties they are expected to study. This paper provides a useful starting point towards such a systematization by providing a common model and a shared vocabulary of concepts that places problems of security and privacy into the context of causality, experimentation, and statistical analysis.

Overview: We explore, both empirically and theoretically, how to conduct IFA over black-box systems while avoiding unjustified assumptions. First, in Section II, we build upon traditional IFA by starting with noninterference [22], a standard formalization of information flow. We identify the limited abilities of the analyst in these problems and cast WDUD as a form of analysis between the extremes of white-box program analysis and black-box monitoring. In doing so, we shift IFA from its traditional context of program analysis using white-box models of software to the new context of investigating black-box systems that hide much of their behavior and operate in uncontrolled environments. We thereby highlight an interesting flavor of black-box IFA that lacks prior formal study.

In particular, we focus on WDUD as it is the least understood nontraditional IFA problem. We formalize it in terms of noninterference showing its relationship to IFA. We prove that sound information flow detection is impossible in this setting (Theorem 1).

Motivated by this impossibility result, we look for an alternative statistical approach. To do so, we build upon research on causality [23], strengthening the connection between the two research areas. In Section III, we prove that a system has interference from a high-level user H to a low-level user L in the sense of IFA if and only if inputs of H can have a causal effect on the outputs of L while the other inputs to the system remain fixed (Theorem 3). This connection allows us to appeal to inductive methods employed in experimental science to study IFA. Such methods provide precisely what we need to make high-assurance statistical claims about flows despite our impossibility result. We leverage this observation to approach WDUD with information flow experiments.

Section IV discusses the design of information flow experiments. We show a correspondence between the features of WDUD and the features of a scientific study (Table I). We discuss the limitations and abilities of experiments to find interference. In particular, we explain the difficulty of finding that a single system has interference. We also identify the ability to find that a system and its environment acting together has interference. For example, we find that while we cannot claim Google itself has interference, we can determine that Google and it's ad ecosystem does.

In Section V, we review significance testing as a systematic method of quantifying the degree of certainty that an information flow experiment has observed interference. We conduct pilot studies to explore what assumptions, and therefore statistical analyses, are appropriate for WDUD. We identify permutation testing [24], a method of significance testing, as particularly well suited. In essence, it uses randomization, similarly to security algorithms, to defeat adversaries, making it appropriate for a security setting.

Section VI compares our method to those found in prior WDUD studies. In Section VII, we empirically benchmark our interpretations of some of their approaches with our own WDUD study. This WDUD study is the first to come with a methodology showing its relationship to IFA.

We then provide practical suggestions, summarized in Section VIII, for systematically conducting future WDUD studies. We end by discussing directions for new research that further strengthens the connection between information flow and causality and applies it to other security problems.

Throughout this work, we present our own experiments to illustrate the abstract concepts we present. These results may also be of independent interest to the reader.

A related technical report contains details of experiments and results, formal models, and the proof of each of our theorems [25]. The code and data collected are available at www.cs.cmu.edu/~mtschant/ife.

Prior Work: Three of the authors augmented our method and applied it to run information flow experiments on Google [26]. That paper does not claim the results we present herein as contributions.

Ruthruff, Elbaum, and Rothermel note the usefulness of experiments for program analysis [27]. Whereas our work focuses on problems where traditional white-box analyses are impossible, their work examines experiments in the more traditional setting where the analyst has control over the system in question. Furthermore, we develop a formalism relating informal flow and causality, provide proofs, and present a statistical analysis.

While we could not find any prior articulation of the formal correspondence between informal flow and causality (our Theorem 3), we are not the first to note such a connection. McLean [20] and Mowbray [21] each proposed a definition of information flow that uses the lack of a causal connection to rule out security violations even if there is a flow of information from the point of view of information theory. Sewell and Vitek provide a "causal type system" for reasoning about information flows in a process calculus [28]. We differ from these works by showing an

equivalence between a standard notion of information flow, noninterference [22], and a standard notation of causality, Pearl's [23], rather than using a notion of causality to adjust an information theoretic notion of information flow. Furthermore, Mowbray's formalism requires white-box access to the system while McLean's only considers temporal ordering as a source of causal knowledge. More importantly, they use causality to handle problematic edge cases in their formalisms whereas we reduce interference to causality so that we may apply standard methods from experimental science to IFA.

Our identification of WDUD as an interesting problem for IFA was inspired by prior WDUD studies. Sweeney uses a method similar to ours to study how search ads are affected by search terms [5]. Our work provides a formal justification of her method in terms of information flows. Other prior studies either approach the problem with non-statistical analyses [2]–[4], [6] or make assumptions that our experiments show unlikely to hold in the setting we study [7]–[9]. Section VI details these works.

We draw on works from experimental design and statistics, whose discussion we defer until the point of use.

II. Information Flow Analysis

In this section, we discuss prior work on information flow analysis starting with noninterference, a formalization of information flows. We next discuss the analyses used in prior work to determine whether a flow of information exists. We present them systematically by the capabilities they require of the analyst. We end by discussing the capabilities of the analyst in our motivating applications, and WDUD in particular, how prior analyses are inappropriate given these capabilities, and the inherent limitations of these capabilities.

A. Noninterference

Goguen and Meseguer introduced *noninterference* to formalize when a sensitive input to a system with multiple users is protected from untrusted users of that system [22]. Intuitively, noninterference requires the system to behave identically from the perspective of untrusted users regardless of any sensitive inputs to the system.

As did they, we will define noninterference in terms of a synchronous finite-state Moore machine. The inputs that the system accepts are tuples where each component represents the input received on a different input channel. Similarly, our outputs are tuples representing the output sent on each output channel. For simplicity, we will assume that the machine has only two input channels and two output channels, but all results generalize to any finite number of channels.

We partition the four channels into H and L with each containing one input and one output channel. Typically, H corresponds to all channels to and from high-level users, and L to all channels to or from low-level users. The high-level information might be private or sensitive information that should not be mixed with public information, denoted by L. In the area of taint analysis, the roles are reversed in that the tainted information is untrusted and should not be mixed with trusted information on the trusted channel. However, either way, the goal is the same: keep information on the input channel of H from reaching the output channel of L.

We will often have a single user using channels of both sets since we are concerned with not only to whom information flows but also under what contexts. To this end, we interpret *channel* rather broadly to include virtual channels created by multiplexing, such as a field of an HTML form or the ad container of a web page. We also allow each channel's input/output to be a null message indicating no new input/output.

A system q consumes a sequence $\vec{\imath}$ of input pairs where each pair contains an input for the high and the low input channels. We write $q(\vec{\imath})$ for the output sequence $\vec{o}$ that q would produce upon receiving $\vec{\imath}$ as input where output sequences are defined as a sequence of pairs of high and low outputs.

For an input sequence $\vec{\imath}$, let $\lfloor \vec{\imath} \downarrow L \rfloor$ denote the sequence of low-level inputs that results from removing the high-level inputs from each pair of $\vec{\imath}$. That is, it "purges" all high-level inputs. We define $\lfloor \vec{o} \downarrow L \rfloor$ similarly for output sequences.

Definition 1 (Noninterference)**.** *A system q has* noninterference *from L to H iff for all input sequences $\vec{\imath}_1$ and $\vec{\imath}_2$,*

$$\lfloor \vec{\imath}_1 \downarrow L \rfloor = \lfloor \vec{\imath}_2 \downarrow L \rfloor \textit{ implies } \lfloor q(\vec{\imath}_1) \downarrow L \rfloor = \lfloor q(\vec{\imath}_2) \downarrow L \rfloor$$

Intuitively, if inputs only differ in high-level inputs, then the system will provide the same low-level outputs.

To handle systems with probabilistic transitions, we will employ a probabilistic version of noninterference similar to the previously defined *P-restrictiveness* [29] and *probabilistic nondeduciblity on strategies* [30]. To define it, we let $Q(\vec{\imath})$ denote a probability distribution over output sequences given the input $\vec{\imath}$, a concept that can be made formal given the probabilistic transitions of the machine [30]. We define $\lfloor Q(\vec{\imath}) \downarrow L \rfloor$ to be the distribution μ over sequences $\vec{\ell}$ of low-level outputs such that $\mu(\vec{\ell}) = \sum_{\vec{o} \text{ s.t. } \lfloor \vec{o} \downarrow L \rfloor = \vec{\ell}} Q(\vec{\imath})(\vec{o})$. Probabilistic Noninterference compares such distributions for equality.

Definition 2 (Probabilistic Noninterference)**.** *A system Q has* probabilistic noninterference *from L to H iff for all input sequences $\vec{\imath}_1$ and $\vec{\imath}_2$,*

$$\lfloor \vec{\imath}_1 \downarrow L \rfloor = \lfloor \vec{\imath}_2 \downarrow L \rfloor \textit{ implies } \lfloor Q(\vec{\imath}_1) \downarrow L \rfloor = \lfloor Q(\vec{\imath}_2) \downarrow L \rfloor$$

B. Analysis

Information flow analysis (IFA) is a set of techniques to determine whether a system has noninterference (or similar properties) for interesting sets H and L. Examples include analyses employing type systems [10], [31], model checking

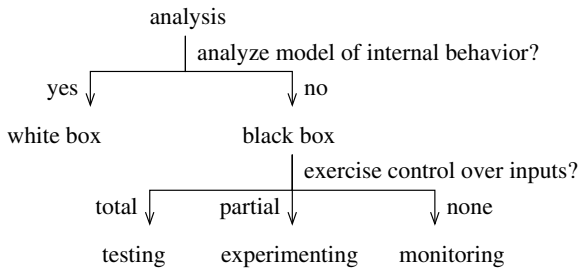

Figure 1. Taxonomy of analyses

of code [32], or dynamic approaches that instrument the code running the system to track values carrying sensitive information (e.g., [33]–[36]).

The above methods are inappropriate for WDUD since they require *white-box* access to the program. That is, the analyst must be able to study and/or modify the code. In our applications, the analyst must treat the program as a *black box*. That is, the analyst can only study the I/O behavior of the program and not its internal structure. Black-box analyses vary based on how much access they require to the system in question. Figure 1 shows a taxonomy of analyses.

Numerous black-box analyses for detecting information flows operate by running the program rather than analyzing its code [37]–[41]. They run the program multiple times with varying inputs to detect changes in output that imply interference. However, these black-box analyses continue to require access to the internal structure of the program even if they do not analyze that structure. For example, the analysis of Yumerefendi et al. requires the binary of a program to copy it into a virtual machine for producing I/O traces [37]. In theory, such black-box analyses could be modified to not require any access to code by completely controlling the environment in which the program executes. To do so, the analyst would run a single copy of the program and reset its environment to simulate having multiple copies of the system. We call this form of black-box analysis, with total control over the system, *testing* as it is the setting typical to software testing.

Testing will not work for our applications. For example, in the setting of WDUD, the analyst cannot reset and run the program multiple times since the analyst has only limited interactions with the program over a network. Thus, it cannot force the program into the same initial environment to reset it. Furthermore, unlike a program, Google's ad *system* is stateful and, thus, modifying its environment alone would be insufficient to reset it. In this setting, the analyst must analyze the *system* as it runs, not a program whose environment the analyst can change at will.

At the opposite extreme of black-box analysis is *monitoring*, which passively observes the execution of a system. While some monitors are too powerful by being able to observe the internal state of the running system (e.g. [42]), others match our needs in that the analyst only has access to a subset of the program's outputs (e.g., [43]). However, all monitors are too weak since they cannot provide inputs to the system as our application analysts can. We need a form of black-box analysis between the extremes of testing and monitoring.

Thus, we find no prior work on IFA that corresponds to the capabilities of the analyst in WDUD or our other motivating applications.

C. Information Flow Experiments

Unlike the primary motivation of traditional IFA, developing programs with Mandatory Access Controls (MAC), our motivating examples involve situations in which the analyst and the system in question are not aligned. Thus, the information available to the analyst is much more limited than in the traditional security setting. In particular, the analyst

1) has no model of or access to the program running the system,
2) cannot observe or directly control the internal states of the system,
3) has limited control over and knowledge of the environment of the system,
4) can observe a subset of the system's outputs, and
5) has control over a subset of the inputs to the system.

We will call performing IFA in this setting *experimenting*. Experiments are an interactive extension of a limited form of execution monitoring that allows analyst inputs but limits the analyst to only observing a subset of system I/O.

Prior work shows that no monitor can detect information flows [42], [44], [45]. We argue that experiments, with their additional ability to control some inputs to the system, do not improve upon this situation. In particular, we prove that no non-degenerate experiment can be sound for interference or for noninterference, even on deterministic systems. (Although, we will later show that experiments do enable statistical analyses with probabilistic soundness properties.)

Experiment Model: We model an experiment as a pair $\langle \vec{\imath}, d \rangle$ where $\vec{\imath}$ is an input sequence and d is a *decision function* from the set of output sequences to $\{\mathsf{ni}, ?, \mathsf{in}\}$. $\vec{\imath}$ represents the sequence of inputs that the analyst supplies to the system in question and d represents how the analyst goes from the sequence of resulting outputs to either the conclusion that the system q has interference (in), has noninterference (ni), or that she does not know (?), all for a fixed H and L. The result of the experiment $\langle \vec{\imath}, d \rangle$ on the deterministic system q is $d(q(\vec{\imath}))$. We allow the analyst just one I/O sequence since the analyst cannot restart the system, which would include resetting its hard drives, clocks, etc. to their initial states. The analyst can embed into its single sequence multiple subsequences each corresponding to a run of a program on the system.

Adversary Model: The system in question q might be under the control of an adversary trying to trick the analyst as to whether the system has interference. We model the adversary as being able to select any automaton for the system q. In essence, the following theorems show that for any experiment, the adversary can select a system that fools the analyst.

To prove the unsoundness of black-box experiments for interference, we consider an arbitrary system q for which an experiment returns a positive result indicating interference. In our setting, the experiment must base its decision solely upon its interactions with the system. Thus, it will return the same positive result for a system q_{N} that always produces the same outputs as q did irrespective of its inputs. Since q_{N} always produces these outputs, it has noninterference making the positive result false.

Theorem 1 (Unsoundness for Interference)**.** *For all experiments $\langle \vec{\imath}, d \rangle$, if there exists a system q such that $d(q(\vec{\imath})) = \mathsf{in}$, then there exists a system q_{N} with noninterference from H to L such that $d(q_{\mathsf{N}}(\vec{\imath})) = \mathsf{in}$.*

The argument for noninterference is symmetric, but requires that interference is possible given the system's input and output space. That is, the system must have at least two high inputs and two low outputs.

Theorem 2 (Unsoundness for Noninterference)**.** *If H has two inputs and L has two outputs, then for all experiments $\langle \vec{\imath}, d \rangle$, if there exists a system q such that $d(q(\vec{\imath})) = \mathsf{ni}$, then there exists a system q_{I} with interference from H to L such that $d(q_{\mathrm{I}}(\vec{\imath})) = \mathsf{ni}$.*

Note that these theorems hold even if the analyst can observe every input in H and L making the above shift of focus to the composite system of Google operating in its environment unsuccessful. However, as we will later see, we can probabilistically handle the lack of total internal control of the composite system using statistical techniques. Since we can never be sure whether we have started a particular sequence of inputs from the same initial state as another sequence, we use many instances of each sequence instead of one for each. Intuitively, if the outputs for one group of inputs are consistently different from outputs for the other group of inputs, then it is likely that the difference is introduced by the difference between the groups instead of from the initial states differing. We formalize this idea to present a probabilistically sound method of detecting interference. We leave detecting noninterference to future work.

III. Causality

In this section, we discuss a formal notion of causality motivated by the studies of the natural sciences. We then prove that noninterference corresponds to a lack of a causal effect. This result allows us to repose WDUD as a problem of statistical inference from experimental data using causal reasoning.

A. Background

Let us start with a simple example. A scientist might like to determine whether a chemical causes cancer in mice. More formally, she is interested in whether the value of the *experimental factor* X, recording whether the mouse ingests the chemical, causes an effect to a *response variable* Y, an indicator of mouse cancer, holding all other factors (possible causes) constant.

Pearl [23] provides a formalization of *effect* using *structural equation models* (SEMs), a formalism widely used in the sciences (e.g., [46]). A probabilistic SEM $M = \langle \mathcal{V}_{\mathsf{en}}, \mathcal{V}_{\mathsf{ex}}, \mathcal{E}, \mathcal{P} \rangle$ includes a set of *variables* partitioned into *endogenous* (or dependent) variables $\mathcal{V}_{\mathsf{en}}$ and *exogenous* (or independent) variables $\mathcal{V}_{\mathsf{ex}}$. M also includes in $\mathcal{E}$, for each endogenous variable V, a *structural equation* $V := F_V(\vec{V})$ where $\vec{V}$ is a list of other variables other than V and F_V is a possibly randomized function. A structural equation is directional like variable assignments in programming languages. Each exogenous variable is defined by a probability distribution given by $\mathcal{P}$. Thus, every variable is a random variable defined in terms of a probability distribution or a function of them.

Let M be an SEM, X be an endogenous variable of M, and x be a value that X can take on. Pearl defines the *sub-model* $M[X{:=}x]$ to be the SEM that results from replacing the equation $X := F_X(\vec{V})$ in $\mathcal{E}$ with the equation $X := x$. The sub-model $M[X{:=}x]$ shows the *effect* of setting X to x. Let Y be an endogenous variable called the *response variable*. We define *effect* in a manner similar to Pearl [23].

Definition 3 (Effect)**.** *The experimental factor X has an* effect *on Y given $Z := z$ iff there exists x_1 and x_2 such that the probability distribution of Y in $M[X{:=}x_1][Z{:=}z]$ is not equal to its distribution in $M[X{:=}x_2][Z{:=}z]$.*

Intuitively, there is an effect if $F_Y(x_1, \vec{V}) \neq F_Y(x_2, \vec{V})$ where $\vec{V}$ are the random variables other than X and Y.

B. The Relationship of Interference and Causality

Intuitively, interference is an effect from a high-level input to a low-level output. Noninterference corresponds to lack of an effect, which Pearl calls *causal irrelevance* [23]. We can make this connection formal by providing a conversion from a probabilistic system to an SEM.

Given a probabilistic Moore Machine Q, we define a SEM M_Q. Intuitively, it contains endogenous variables for each input and output and exogenous variables for each user. In more detail, for each time t, M_Q contains the endogenous variables HI_t. It also contains HO_t for high outputs, LI_t for low inputs, and LO_t for low outputs, all at the time t. M_Q also has exogenous variables HU_t and LU_t that represent the behavior of high and low users of the system at time t.

The behavior of Q provides functions $F_{\mathsf{lo},t}$ defining the low output at time t in terms of the previous and current inputs. The output may depend upon previous inputs via a variable S_t representing the state of the system. Similar functions exist for the other variables. For example, the function $F_{\mathsf{s},t}$ for updating the state is determined by the transition function of Q.

The following lemma shows that Q and M_Q are equivalent. To state it, we use $\vec{V}^t$ to denote the vector holding those variables V_t with an index of t or less (in order). We let I^t represent a similar vector of input variables combining HI^t and LI^t. We use $\vec{V}^t = \vec{v}$ as shorthand for $\bigwedge_{j=1}^{t} \vec{V}[j] = \vec{v}[j]$.

Lemma 1. *For all Q, t, $\vec{\imath}$, and $\vec{\mathsf{lo}}$ of lengths t and $t+1$, respectively, $\mathcal{P}(\vec{\mathsf{LO}}^{t+1}{=}\vec{\mathsf{lo}} \mid \mathsf{do}(\vec{I}^t{:=}\vec{\imath})) = \lfloor Q(\vec{\imath}){\downarrow}L \rfloor(\vec{\mathsf{lo}})$.*

The key theorem follows from Lemma 1 and the properties of SEMs and interference:

Theorem 3. *Q has probabilistic interference iff there exists low inputs $\vec{\mathsf{li}}$ of length t such that $\vec{\mathsf{HI}}^t$ has an effect on $\vec{\mathsf{LO}}^t$ given $\vec{\mathsf{LI}}^t := \vec{\mathsf{li}}$ in M_Q.*

Notice that Theorem 3 requires that the low-level inputs to the system in question be fixed to a set value $\vec{\mathsf{li}}$. This requirement is a reflection of how noninterference only requires that low-level outputs be equal when low-level inputs are equal (Definition 1).

IV. Experimental Design

Having reduced the problem of information flow experiments to that of checking for effects, we can employ the checking method often used in empirical sciences, randomized controlled experiments. However, doing so requires mapping the features of WDUD and other black-box IFA problems into the standard terms of experimental design. Furthermore, it requires scoping the experiment to be within the limited abilities of the analyst. In particular, we must respect the requirement of Theorem 3 that the analyst be able to fix all low-level inputs. We discuss each of these issues before turning to the actual running of the experiment.

A. The Setup of Experiments

A randomized controlled experiment randomly assigns each *experimental unit*, such as a mouse, to either a *control* or an *experimental* treatment. The treatment determines the value of the unit's *experimental factor*, which maps to the changed variable X in Definition 3. The experimenter holds other factors under her control constant to isolate the effect of the treatment. These factors map to Z in Definition 3. The experimenter measures a *response*, some feature, of each unit. The experimenter attempts to determine whether the treatments have an effect on the measured responses.

For example, consider a WDUD study to determine whether a pattern of behavior, or profile, affects the ads that Google shows to a user. Table I summarizes how to view it and an archetypal cancer study as experiments.

General Terms	Chemical Study	Behavioral Marketing
natural process	cancer	marketing
population of units	mice	browser instances
experimental factor	diet	visitor behavior
treatments	chemical or placebo	behavioral profiles
constant factors	water allowance	IP address etc.
noise factors	age, weight, etc.	other users, advertisers
response variables	tumor count	sequences of ads
effect	carcinogenic	use of data

Table I
General Terminology and Two Instances of Experimental Science. In the chemical study, a scientist studies whether a chemical is carcinogenic when added to the diet of mice. In a behavioral marketing study, the scientist studies whether changes in visitor behavior causes changes in ads.

In the case of WDUD, the natural experimental unit might appear to be Google. However, since a randomized controlled experiment requires multiple experimental units and there is just one Google, we must select some subsets of interactions with Google as the experimental units. Since one of the major goals of WDUD is to determine the nature of Google's behavioral tracking of people, interactions with Google at the granularity of people could be an appropriate experimental unit. However, since we desire automated studies, we substitute separate automated browser instances for actual people. In particular, we can use multiple browser instances with separate caches and cookies to simulate multiple users interacting with the web tracker. We can apply treatments to browsers by having them controlled by different scripts that automate different behaviors.

The treatments are various behavioral profiles that the analyst is interested in comparing. The constant factors can include anything the analyst can control: the IP address, the browser used, the time of day, etc. The response may be the ads shown to the simulated browser.

B. Scoping the System

Properly scoping an the experiment for WDUD is particularly important. Suppose in the above example, the system in question is Google. Since the profile of the user is of interest, it dictates the high-level inputs. Since every input must be either low-level or high-level, all inputs not determined by the profile are low-level. These low-level inputs include some that the analyst cannot observe or control, such as inputs from advertisers to Google. However, Theorem 3 requires that the all the low-level variables remain fixed. That is, to use Theorem 3, the analyst must select the system and its inputs so that she can ensure that the low-level inputs are fixed.

The analyst must shrink the set of low-level inputs to just those that she can fix. One means of achieving this goal is to consider more inputs high-level, but if the inputs converted to be high-level are already known to determine the ads shown (such as inputs from advertisers), then the analysis would be of little interest. Another means would be to alter Google so that it no longer accepts such inputs, but the analyst does not have such control over Google. However, the analyst does have control over which system she studies. Rather than study Google in isolation, she could study the composite system of Google and the advertisers operating in parallel. By doing so, she converts the uncontrolled low-level inputs to Google from the advertisers into internal messages of the composite system, which are irrelevant to whether interference occurs.

The practical consequences of these limitations for WDUD is that we cannot determine that Google has interference on its own. Rather, we can only determine that Google operating in its environment has interference. That is, we can determine that the composite system consisting of Google and the other systems making up the ad ecosystem has interference.

This limitation means that we cannot explain how observed interference occurs. Upon seeing interference, one explanation is that Google directly used the information in question to select ads. However, it is also possible that Google shared the information with an advertiser that used the information to change its bids, which, in turn, caused Google to change its ads. If the output to the advertiser is low-level, then Google itself does not have interference in the second case.

Nevertheless, we know that some part of the ad ecosystem used the information. This finding can be useful in its own right if one is interested in the complete process of how ads are selected. It could also justify a white-box investigation by either internal auditors or external regulators who may compel internal access.

Lastly, note this scoping does not enable sound nor complete analyses of the composite system: Theorems 1 and 2 continue to apply since they do not require non-empty sets of low-level inputs. The analyst's continued inability to observe the internal state of the system means that the analyst must still employ statistical analyses.

C. Running the Experiment

With the system properly scoped, we run such a randomized controlled experiment as follows:

1) Assign each browser either an experimental or control profile at random.
2) Each browser instance simulates those profiles by interacting with webpages.
3) Each browser instance collects ads from (possibly other) webpages.
4) Compare the collected ads from browsers with one profile to browsers with the other profile.

In more detail, the analyst prepares a vector $\vec{x}$ with a length equal to the number of units that hold treatment values. Typically, half will be control treatments and half experimental treatments. She randomly assigns each experimental unit k to an index i_k of $\vec{x}$ so that no unit is assigned the same index. For each k, she then applies the treatment in the i_kth slot of $\vec{x}$ to the unit k, which in our setting implies providing inputs corresponding to a profile. Units assigned the same treatment form a *group*.

Groups may vary due to *noise factors*, variations among the experimental units other than those from the application of treatments. Proper randomization over larger sample sizes makes negligible the probability that the groups vary in a systematic manner. If the analyst also ensures that no other systematic differences are introduced to the groups after the application of the treatment, the units will not systematically differ between the groups under the *null hypothesis* that the treatment has no effect. Thus, any difference in responses that consistently shows up in one group but not the other can only be explained by chance under the null hypothesis. If given the sample size, this chance is small, then the analyst can reject the null hypothesis as unlikely, providing probabilistic evidence of a causal relationship, which we quantify in the next section. (See [47] for a more detailed review of these concepts.)

V. Statistical Analysis

To quantify the probability that the collected responses could appear to show a flow of information due to a chance occurrence, we use *significance testing* [48]. A *statistical test* of the data provides a *p-value*, the probability of seeing results at least as extreme as the observed data under the assumption that the null hypothesis is true. A small p-value implies that the data is unlikely under the null hypothesis. Typically, scientists are comfortable rejecting the null hypothesis if the p-value is below a threshold of 0.05 or 0.01 depending on the field. Rejecting the null hypothesis makes the alternative hypothesis that there is an effect more plausible. In our case, using significance testing requires selecting a test of independence. We discuss the process of selecting one and detail the one we have selected, permutation testing.

A. Selecting a Test of Independence and Pilot Studies

Some tests of independence require assumptions about the system in question. These assumptions enable powerful statistical techniques, which in some cases allow smaller sample sizes or more detailed characterizations of a research finding.

Parametric tests assume that the behavior of the system in question is drawn from some known family of distributions with a small number of unknown parameters. Another

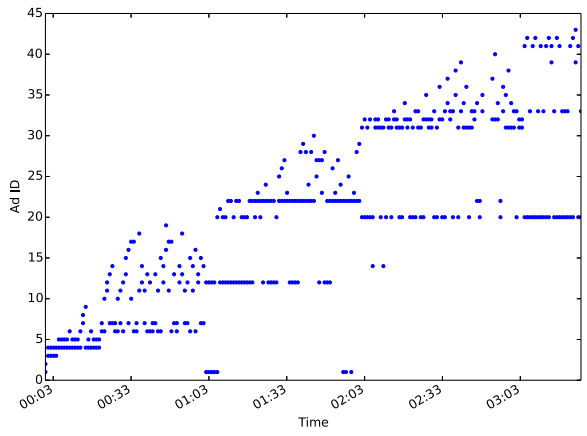

Figure 2. Ads collected from the first browser instance to visit the Chicago Tribune. The time interval for collection was one minute. The x-axis is time measured in hh:mm. The y-axes ranges over unique ads ordered by the time at which the instance first observed it in the experiment.

common assumption is that, under the null hypothesis, the responses of the experimental units are independent of one another and identically distributed (i.i.d.).

Some statistical analyses require that giving or withholding a treatment from one unit will not have an effect upon the other units (e.g., [49, p. 19]), that is, the absence of *cross-unit effects*.

Given our understanding of how ad networks operate, these assumptions appear suspicious. The complex behavior of ad networks makes selecting a family of distributions to model one difficult. The fact that budgets control the number of ad impressions creates the possibility that one browser instance receiving an ad might decrease the probability that another receives it, invalidating the i.i.d. assumption. Any choice of experimental unit other than all of Google, which leads to a sample size of one, will possibly exhibit cross-unit effects by virtue of units being multiplexed onto a single system.

To empirically explore how reasonable these assumptions are in our setting, we conducted two pilot studies, which show them difficult to defend in our setting.

Experiment 1. We collected ads served by Google on a third-party website to understand how they vary over time. Following Balebako et al.'s study [3], we used the Breaking News page of the Chicago Tribune (http://www.chicagotribune.com/news/local/breaking/).

To collect ads, we simultaneously started two browser instances, and collected the ads served by Google on the webpage. Each instance reloaded the web page 200 times, with a one minute interval between successive reloads.

Figure 2 shows a temporal plot of the ads served to one of these instances. The plots suggest that each instance received certain kinds of ads for a period of time, before being switched to receiving a different kind, which implies that *ads are not identically distributed* across time. This pattern held using other intervals for reloads.

One explanation for this behavior is that Google associated users with various ad pools switching users from pool to pool over time. While hierarchical families of parametric models could capture this behavior, we are not comfortable making such an assumption and the resulting models would be more complex than those typically used in parametric tests. Thus, *parametric tests would employ models of low confidence*.

Our results do not mean that one could not reverse engineer enough of Google to find an appropriate model. However, they suggest that such reserve engineering would be difficult. Furthermore, it runs against the spirit of performing black-box information flow analysis.

Shortly after we conducted these experiments, the Chicago Tribune stopped hosting text ads from Google. Thus, when we later wanted to replicate the study, we instead looked at the Times of India, the BBC, and Fox News. The ads continued to appear to violate the i.i.d. assumption with some ads being shown over and over again in streaks. However, the binning behavior was gone. This difference in behavior suggests that any success at reverse engineering Google may be specific to a webpage or short lived as Google changes its behavior.

We carried out this and all other experiments using Python bindings for Selenium WebDriver, which is a browser automation framework. A test browser instance launched by Selenium uses a temporary folder that can be accessed only by the process creating it. So, two browser instances launched by different processes do not share cookies, cache, or other browsing data. All our tests were carried out with the Firefox browser. When observing Google's behavior, we first "opted-in" to receive interest-based Google Ads across the web on every test instance. This placed a Doubleclick cookie on the browser instance. No ads were clicked in an automated fashion throughout any experiment. □

Experiment 2. We studied whether multiple browser instances running in parallel affect one another. We compared the ads collected from a browser instance running alone to the ads collected by an instance running with seven additional browser instances each collecting ads from the same page.

A primary browser instance would first establish an interest in cars by visiting car-related websites. We selected car-related sites by collecting, before the experiment, the top 10 websites returned by Google when queried with the search terms "BMW buy", "Audi purchase", "new cars", "local car dealers", "autos and vehicles", "cadillac prices", and "best limousines". After manifesting this interest in cars, the instance would collect text ads served by Google on the International Homepage of Times of India (http://timesofindia.

indiatimes.com/international-home). We attempted to reload the collection page 10 times, but occasionally it would time out. Each successful reload would have 5 text ads, yielding as many as 50 ads.

Our experiment repeated this round of interest manifestation and ad collection 10 times using a new primary browser instance during each round. We randomly selected 5 of the rounds to include seven additional browsers. When the additional browsers were present, three of them performed the same actions as the primary one. The other four would wait doing nothing instead of visiting the car-related websites and then went on to collecting ads after waiting. All instances would start collecting ads at the same time.

The experiment showed that the primary browsers ran in isolation would receive a more diverse set of ads than those running in parallel with other browsers. We repeated the experiment four times (twice using 20 rounds) and found this pattern each time:

Rounds	Unique ads in isolation	Unique ads in parallel
10	37	25
10	46	33
20	58	47
20	57	52

The presence of this pattern leads us to believe that *cross-unit effects between browser instances exist*. While a statistical test could report whether the observed effect is significant, doing so would inappropriately shift the burden of proof: if a scientist would like to use a statistical analysis that requires an absence of cross-unit effects, then the onus is on her to justify the absence.

This experiment also leads us to suspect that *browser instances are not identically distributed*. In particular, the nature of the cross-unit effects suggested by the experiment raises the possibility that the one browser receiving an ad might decrease the probability for another browser, leading to non-identical distributions. □

Given the results of these experiments, for information flow experiments, we find each of these assumptions to be suspect: parametric models, i.i.d. responses, and the absence of cross-unit effects. Any work employing these assumptions must take care to justify their use in their particular experimental design and setting with pilot studies. Believing that these assumptions do not hold for many such experiments, we instead choose to focus on statistical analyses that do not require making such assumptions.

B. The Permutation Test

Let us look at selecting a statistical test from the angle of security. In our setting, the system in question, not the analyst, is the adversary. From this angle, the pilot studies are reflections of the adversary's ability to violate most assumptions an analyst might wish to make about it. In a security setting, one of the few assumptions safe for the analyst to make is that the adversary cannot guess the (pseudo-)random numbers she generates. Indeed, selecting a random key is the core of many security algorithms, such as encryption.

With the security properties of randomization in mind, we should adopt a statistical test that leverages randomization rather than the types of assumptions more typically seen in statistics. Fortunately, *permutation tests* (e.g., [24]), also known as *randomization tests*, uses randomization to allow cross-unit interactions [50] and non-i.i.d. responses.

At the core of a permutation test is a *test statistic* s. A test statistic is a function from the data, represented as a vector of responses, to a number. The vector of responses $\vec{y}$ has one response for each experimental unit. The vector must be ordered by the random indices i_k used to assign each unit k a treatment from the treatment vector $\vec{x}$ prepared during the experiment. Thus, the kth entry of $\vec{y}$ received the treatment at the kth entry of $\vec{x}$. In particular, s could use the first n components of the data vector as the results of the experimental group and the remaining m as the results for the control group where the groups have n and m units, respectively.

For example, consider an experiment on whether visiting car-related websites impacts the ads one sees. In it, the experimental group visits such websites while the control group does not. The analyst could use a keyword-based test statistic s_{kw}, which looks at the number of ads that each instance received containing the keywords "bmw", "audi", "car", "vehicle", "automobile", "cadillac", and "limo", words whose presence we believe to be indicative of an instance being in the experimental group. Let the value of s_{kw} be the number of ads that contained any of the keywords amongst the experimental group less the number in the control group. Intuitively, a small value would suggest no noteworthy difference between the groups whereas a large value would indicate that the experimental group saw more car ads as a result of visiting car-related websites.

To make this intuition formal, we must quantify "small" and "large" values. Since the scientist is allowed to pick any function s from response vectors to numbers for the test statistic, the permutation test needs to gauge whether an observed data vector $\vec{y}$ produces a large value with respect to s. To do so, it compares the value of $s(\vec{y})$ to the value of $s(\pi(\vec{y}))$ for every permutation π of $\vec{y}$. Intuitively, this permuting mixes the treatment groups together and compares the observed value of s to its value for these arbitrary groupings.

The significance of these comparisons is that under the null hypothesis of independence (noninterference), the groups should have remained exchangeable after treatment and there is no reason to expect $s(\vec{y})$ to differ in value from $s(\pi(\vec{y}))$. Thus, we would expect to see at least half of the

comparisons succeed. Thus, we call a permutation π such that $s(\vec{y}) \leq s(\pi(\vec{y}))$ fails to hold a *rejecting permutation* since too many rejecting permutations leads to rejecting the null hypothesis.

Formally, the value produced by a (one-tailed signed) permutation test given observed responses $\vec{y}$ and a test statistic s is

$$\mathsf{pt}(s, \vec{y}) = \frac{1}{|\vec{y}|!} \sum_{\pi \in \Pi(|\vec{y}|)} I[s(\vec{y}) \leq s(\pi(\vec{y}))] \qquad (1)$$

where $I[\cdot]$ returns 1 if its argument is true and 0 otherwise, $|\vec{y}|$ is the length of $\vec{y}$ (i.e., the sample size), and $\Pi(|\vec{y}|)$ is the set of all permutations of $|\vec{y}|$ elements, of which there are $|\vec{y}|!$.

Recall that under significance testing, a p-value is the probability of seeing results at least as extreme as the observed data under the assumption that the null hypothesis is true. $\mathsf{pt}(s, \vec{y})$ is a (one-tailed) p-value using s and $\leq$ to define *at least as extreme as* in the definition of p-value. To see this, recall that the null hypothesis H_0 is that the treatments have no effect. Thus, since the order of the responses in $\vec{y}$ is by treatment, which should not matter under H_0, and otherwise random, any permutation of them would be equally likely under H_0. Thus,

$$\mathsf{pt}(s, \vec{y}) = \sum_{\pi \in \Pi(|\vec{y}|): s(\vec{y}) \leq s(\pi(\vec{y}))} \Pr[\vec{Y} = \vec{y} \,|\, H_0] \qquad (2)$$

matches the definition of a p-value. One could use other definitions of *as extreme as* by replacing the $\leq$ in (1) and (2) by $\geq$ or by comparing the absolute values of $s(\vec{y})$ and $s(\pi(\vec{y}))$ to check for extremism in both directions (a two-tailed test).

Good discusses using sampling to make the computation of $\mathsf{pt}(s, \vec{y})$ tractable for large $\vec{y}$ [24]. Greenland provides detailed justification of using permutation tests to infer causation [51].

C. Discussion

The above method avoids some pitfalls. Most fundamentally, we use a statistical analysis whose assumptions matches those of our experimental design. Assumptions required by many statistical analyses appear unjustifiable in our setting.

Remark 1. *The permutation test provides a method of determining whether a system has interference that is probabilistically sound to a degree quantified by the p-value.*

Our use of randomization implies that many factors that could be confounding factors in an unrandomized design become noise in our design (e.g., [24]). While such noise may require us to use a large sample size to find an effect, it does not affect the soundness of our analysis. We expect our methodology to suggest that an effect exists when one does not with a probability equal to or less than the p-value.

Remark 2. *The permutation test is not sound for finding noninterference nor complete for finding interference.*

Our method might fail to detect some use of information. For example, the web service's behavior might vary by some feature not measured by the test statistic.

Furthermore, we do not claim that results generalize beyond the setting of the experiment. To do so, our method may be combined with random sampling and methods to ensure that the observed system does not attempt to evade the study.

Lastly, we do not claim that our method shows how the system in question uses information internally. Any observed effect may be the result of complex interactions between the system and other ones in its environment. In particular, as discussed in Section II-C, our method finds interference not within the system in isolation, but rather for the system operating in its environment.

VI. Prior Studies of WDUD

Our work was inspired by prior WDUD studies. Table II provides an overview of those using other methods. The first four studies used non-statistical analyses, Sweeney's uses a method similar to ours, and the last four use statistical analyses under assumptions that appear not to hold in the setting we study. Additionally, a recent study under submission has adopted our method [26].

The method of Guha et al. [2], which Balebako et al. adopt to study the effectiveness of web privacy tools [3], uses three browser instances. Two of them receive the same treatment and can be thought of as controls. The third receives some experimental treatment. They collect the ads Google serves each of them, which they compare using a similarity metric. Based on experimental performance, they decided to use one that only looks at the URL displayed in each ad. For each instance, they perform multiple page reloads and record the number of page reloads for which each displayed URL appears. From these counts, they construct a vector for each unit where the ith component of the vector contains the logarithm of the number of reloads during which the ith ad appears. To compare runs, they compare the vectors resulting from the instances using the cosine similarity of the vectors.

More formally, their similarity metric is $\mathsf{sim}(\vec{v}, \vec{w}) = \mathrm{coss}(\ln^*(\vec{v}), \ln^*(\vec{w}))$ where $\vec{v}$ and $\vec{w}$ are vectors that record the number of page reloads during which each displayed URL ad appears, $\ln^*$ applies a logarithm to each component of a vector, and coss computes the cosine similarity of two vectors. They conclude that a flow of information is likely if $\mathsf{sim}(\vec{v}_{\mathsf{c1}}, \vec{v}_{\mathsf{c2}})$ is much larger than $\mathsf{sim}(\vec{v}_{\mathsf{c1}}, \vec{v}_{\mathsf{e}})$ where $\vec{v}_{\mathsf{c1}}$ and $\vec{v}_{\mathsf{c2}}$ are the responses from the two control instances and $\vec{v}_{\mathsf{e}}$ is the response from the experimental instance.

Wills and Tatar also studied how Google selects ads by posing as various sorts of website visitors [4]. They drew conclusions in two ways. The first was similar to Guha

Work	Venue	Year	Approach	Limitations/Assumptions
Guha et al. [2]	IMC	2010	cosine similarity	lacks test of statistical significance
Balebako et al. [3]	Web 2.0 Sec. & Priv.	2012	cosine similarity	lacks test of statistical significance
Wills and Tatar [4]	WPES	2012	manual examination	lacks test of statistical significance
Sweeney [5]	CACM	2013	randomized χ^2 test over browsers	requires a large sample size
Liu et al. [6]	HotNets	2013	process of elimination	lacks test of statistical significance
Barford et al. [7]	WWW	2014	non-randomized χ^2 test over ads	assumes ads identically distributed
Lécuyer et al. [8]	Usenix Security	2014	parametric model over ads	correlation; assumes ads identically distributed and model
Englehardt et al. [9]	submitted	2014	parametric model over ads	assumes independence between ads

Table II
PRIOR WORKS EXPERIMENTING ON ONLINE MARKETING SYSTEMS WITH OTHER METHODS

et al.'s methodology of looking for differences between the ads seen by various profiles. While their study was conducted by hand and without statistics, they intuitively used the keyword test statistic similar to s_{kw} presented in Section V-B. The second was to observe Google showing them ads that included sensitive information they provided to Google by interacting with a website that uses a Google service.

Liu et al. provide *AdReveal*, a system designed to determine the reasons behind the ads a user sees [6]. They consider three types of ads: (1) ads for particular products that the user has previously expressed interest in, (2) ads based on the context of the website, and (3) ads from behavioral marketing. To check for the first, they check the webpage for Javascript code that sets up re-marketing frameworks. To check for the second, they use machine learning to construct a model that intuitively judges whether an ad and a webpage cover the same topic. They consider ads that do not trigger either of these checks to be behavioral.

None of these studies performed a statistical analyses to show whether or not their results are significant. It remains possible that the differences observed are simply from random variations caused by factors other than behavioral marketing.

Sweeney examined the flow of information from a search field to ads shown alongside the search results [5]. Among other things, she found that, compared to characteristically white names, searching for characteristically black names yielded more ads for InstantCheckmate that were suggestive of the searched name having an arrest record. She randomized the order of names searched [52], which can enable statistical analyses without making questionable assumptions. She used the χ^2 test to analyze her results and found them to be significant.

Under some conditions, the χ^2 test becomes an approximation of the permutation test [53]. With the size of her data set, such approximations become not only accurate, but useful for computational reasons. Indeed, her results are also statistically significant under the permutation test [52].

We believe that our methodology with permutation testing provides a foundation for such approximations by linking them to information flow, especially considering that the traditional justification of the χ^2 test includes an assumption that the experimental units are independent [54], which seems unlikely in some cases (Experiment 2). The permutation test is also more flexible in that it works for any response variable whereas the χ^2 test is only for a single categorical (binary) response from each experimental unit (e.g., browser instance). For example, the χ^2 test cannot be used if the response is the number of times a certain ad appears to each browser since this is not a categorical response.

For each webpage, ad, and user profile, Barford et al. [7] record the number of times that webpage shows that ad to a user with that profile as they crawl the web. Using the χ^2 test, they identify those ads shown on a webpage to some profiles more often than others, suggesting behavioral targeting. However, rather than use randomization for the purposes of the χ^2 test, they counted each ad as an i.i.d. response [55] (cf. Experiments 1 and 2).

Lécuyer et al. present XRay, a tool that looks for correlations between the data that web services have about users and the ads shown to users [8]. While their tool may check many changes to a type of input to determine whether any of them has a correlation with the frequency of a single ad, it does not check for causation, as our method does. Furthermore, they assume identically distributed ads (cf. Experiment 1).

Englehardt et al. study filter bubbles with an analysis that assumes a binomial model and independence between observations [9] (cf. Experiment 2).

VII. COMPARISON OF TEST STATISTICS

Given all the test statistics discussed, one might wonder how they compare. We will empirically compare a subset of the test statistics in our motivating setting of WDUD. However, we caution that our experiment should not be considered definitive since other WDUD problems may result in different results. We recommend that each experiment is preceded by a pilot study to determine the best test(s) for the experiment's needs. For example, we have found

pilot studies useful for selecting distinguishing keywords to search for in ads. Other work discusses automating this process for our method [26].

The following experiment also illustrates our experimental design and statistical analysis. We find that it works as expected by running it in settings where we can be almost certain that targeting either is or is not happening.

Experiment 3. Each run of the experiment involved ten simultaneous browser instances, each of which represents an experimental unit. We used a sample size of ten due to the processing power and RAM restrictions of our server. For each run, the script driving the experiment randomly assigns five of the instances, the experimental group, to receive the treatment of manifesting an interest in cars (a heavily marketed topic). As in Experiment 2, an instance manifests its interest by visiting the top 10 websites returned by Google when queried with certain automobile-related terms: "BMW buy", "Audi purchase", "new cars", "local car dealers", "autos and vehicles", "cadillac prices", and "best limousines". The remaining five instances made up our control group, which remained idle as the experimental group visited the car-related websites. Such idling is needed to remove time as a factor ensuring that the only systematic difference between the two groups was the treatment of visiting car-related websites.

As soon as the experimental group completed visiting the websites, all ten instances began collecting text ads served by Google on the International Homepage of Times of India. As in Experiment 2, each instance attempted to collect 50 text ads by reloading a page of five ads ten times, but page timeouts would occasionally result in an instance getting fewer. We repeated this process for 20 runs with fresh instances to collect 20 sets of data, each containing ads from each of ten instances.

Across all runs of the experiment, we collected 9832 ads with 281 being unique. Instances collected between 40 and 50 ads with two outliers each collecting zero. Both outliers were in the 19th run and in the experimental group. We analyzed the data with multiple test statistics.

First, we used the permutation test with s_{sim}, an extension of Guha et al.'s cosine similarity metric sim for comparing more than two responses (Section VI), as the test statistic [2]. We first aggregate together multiple URL-count vectors by computing the average number of times each URL appeared across the aggregated units. Formally, let $\mathsf{avg}(\vec{u})$ compute the component-wise average of the vectors in $\vec{u}$, a vector of vectors of URL counts. We can then define the test statistic $s_{\mathsf{sim}}(\vec{y})$ as $-\mathsf{sim}(\mathsf{avg}(\vec{y}_{1:n}), \mathsf{avg}(\vec{y}_{n+1:n+m}))$ where $\vec{y}_{a:b}$ is the sub-vector consisting of the entries a though b of $\vec{y}$, the first n responses are from the experimental group, and the next m are those from the control group. We use negation since our permutation test takes a metric of difference, not similarity. Intuitively, the permutation test using the test statistic s_{sim} will compare the *between-group* dis-similarity to the dis-similarity of vectors that mix up the units by a permutation. In aggregate, the dis-similarity of these mixed up vectors provide a view on the global dis-similarity inherit in the system.

Observe that there are $10! > 3$ million different permutations for the ten instances. However, since sim treats the response vector provided to it as two sets (intuitively, the experimental and control groups) many permutations will produce the same value for s_{sim}. To speed up the calculation, we replaced comparing all permutations with comparing all partitions of the responses into equal sized sets of 5, yielding only $\binom{10}{5} = 252$ comparisons.

Second, we tested the statistic s_{kw} inspired by Wills and Tatar [4] and discussed in Section V-B, which looks at the number of ads that each instance received containing a keyword. As with s_{sim}, we have at most 252 unique comparisons to make.

Third, we tested a simplified version of s_{kw}, s_{kw01}, that treats each browser instance as providing a categorical (binary) response that is 1 if it got any number of ads with a keyword and 0 if it got none. Comparing s_{kw} to s_{kw01} illustrates the power of non-categorical responses.

Lastly, we conducted the χ^2 test on a 2×2 contingency table computed from the data from each round. The type of treatment was represented in rows, while the presence or absence of keywords was represented in the columns, using an approach similar to s_{kw01} since the χ^2 test is limited to categorical variables. However, our sample size was not large enough to produce meaningful results from the χ^2 test, which requires that each outcome shows up with a certain minimum frequency. Thus, we did not consider this test further.

For comparison purposes, we re-run the above experiment without having the experimental group manifest any interests. That is, we compared two control groups against one another expecting not to find statistically significant differences.

Table III summarizes the results using the standard $\alpha = 0.05$ cutoff for statistical significance. For the experiment-control setup, in which we do expect to find a difference between the groups, both s_{sim} and s_{kw} reported a positive result 18 out of 20 times whereas the s_{kw01} reported no significant results. For s_{kw}, 13 of the p-values were less than 0.004. They show, with high certainty, that Google and its ad ecosystem has interference from visiting the car related webpages to the ads that Google shows. Furthermore, these results show that s_{kw} is indeed a more powerful test statistic than s_{kw01}, which is limited to categorical responses.

As expected with a theoretical false positive rate of $\alpha = 5\%$ and 20 tests, we found that each of the permutation tests produced one or fewer statistically significant results for the experiments with no difference (control-control and experiment-experiment). We provide no minimal acceptable

Setup	Actual positives	Reported positives		
		s_{sim}	s_{kw}	s_{kw01}
Experiment-control	20	18	18	0
Control-control	0	1	1	1
Experiment-experiment	0	0	1	0

Table III
THE NUMBER OF RUNS OUT OF 20 EXPERIMENTS THAT VARIOUS TESTS CONSIDER TO SHOW INFORMATION USAGE.

number of true positives for the experiment-control setup since significance testing does not guarantee any rate. □

The wide range of tests might tempt one into running more than one test on a data set. However, running multiple tests increases the chance of getting a low p-value for one of them by an unlucky randomization of units rather from an effect. Thus, one cannot look just at the test that produced the lowest p-value. Rather one must report them all or apply a correction for multiple tests such as those for the false discovery rate [56].

VIII. Conclusions and Suggestions

Based upon theoretical results, we reduced finding a flow of information in WDUD and related applications to that of finding an effect using experiments. Based upon empirical observations, we recommended an experimental design and a statistical analysis, based on permutation testing, that is well suited to studying behavioral marketing. This process allows us to convert the abstract principles of experimental design and analysis into concrete suggestions:

1) *Use an appropriate statistical test.* Attempting to shoehorn data into familiar statistics can result in incurring requirements, such as i.i.d. data, that cannot be met in our setting. Fortunately, they are *not* required for permutation testing.
2) *Randomly assign treatments to units.* Randomization provides the justification needed for permutation testing, which allows us to avoid unachievable requirements needed for other statistical tests.
3) *Use domain knowledge gained during pilot studies to select a test statistic.* Finding the correct keywords to examine in ads allowed us to not only get results that were statistically significant, but also intuitive.

While statistical analyses can be intimidating due to their complex requirements, selecting the correct test is liberating by also identifying what conditions the analyst needs not worry about. In addition to not needing i.i.d. data, permutation testing does not require complete control over the environment or a lack of cross-unit effects, none of which are achievable in our setting.

Furthermore, we do not require random samples. Acquiring units by randomly sampling from a more general population will, with high likelihood, provide a representative sample, which allows findings of effects to generalize to the population as a whole. While results need not be general to show that a marketer tracks some behavior, showing that the marketer often does is more interesting. However, given that the marketer could alter its behavior in response to the atypical patterns of access exhibited by experiments, we take comfort in knowing that our findings of information usage will hold even if they do not generalize to marketer's typical behavior. (Also, it would be odd for a marketer to only exhibit questionable behavior to those looking for it.)

In general, information flow experiments allow an analyst to exercise oversight and detect transgressions by an entity not controlled by the analyst and unwilling or unable to provide the analyst complete access to the system. We see this setting becoming ever more common: data lives in the cloud, jobs are outsourced, products licensed, and services replace infrastructure. In each of these cases, a party has ceded control of a resource for efficiency. Nevertheless, each party must ensure that the other abides by their agreement and respect privacy policies while having only limited access to the other. Thus, we envision black-box experimentation for auditing and accountability playing an increasing role in information security, privacy, and society in general.

IX. Future Directions

Demonstrating Noninterference: The permutation test requires that the null hypothesis be that the system has noninterference. Thus, it can only provide a quantitative measure of the evidence *against* noninterference. Conceptually, proving noninterference would require looking at every test statistic under every input sequence. Since examining an infinite set of sequences is impossible, using the scientific method to show that a system has noninterference would require building a theory of the system's operation and then proving noninterference in that theory.

Other Notions of Information Flow and Causation: We examined only one information flow property, a probabilistic noninterference, and one notion of causality, effect. Exploring the many alternatives could tighten the connection between the two fields and further organize each. For example, some definitions of information flow differ from noninterference by being epistemic in nature, that is, they are defined in terms of a change in an agent's state of knowledge (e.g., [57]). We believe that this dichotomy is mirrored in causality by the distinction between causation and association (i.e., correlation). Also, differential privacy [58], being an approximate form of our definition of probabilistic noninterference, is a causal rather than information theoretic notion, which explains why it works for all adversaries regardless of their knowledge.

Formalizing Permutation Testing as IFA: In Sections IV and V, we leverage the relationship of interference and causal effects to use a standard experimental design and statistical analysis to find interference. Future work could

model the experimental design itself as a network of automata. Doing so would allow us to formalize permutation testing in terms of the resulting composite system. Looking at statistical concepts from the fresh angle of IFA could shed light on statistical concerns and yield rigorous automated experiments.

Monitoring and Observational Studies: Passive monitoring in IFA corresponds to observational studies. A wide range of work deals with the cases under which one can infer causation from a correlation learned from an observational study (see, e.g., [23]). Future work can import these results to IFA showing how monitoring could be useful in some cases despite its inherit unsoundness [42], [44], [45].

ACKNOWLEDGMENTS

We thank Augustin Chaintreau, Roxana Geambasu, Qiang Ma, Latanya Sweeney, and Craig E. Wills for providing additional information about their works. We thank Divya Sharma, Arunesh Sinha, and the reviewers of this paper for their helpful comments. This research was supported by the National Science Foundation (NSF) grants CCF0424422, CNS1064688, and CNS1065240. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

REFERENCES

[1] B. Ur, P. G. Leon, L. F. Cranor, R. Shay, and Y. Wang, "Smart, useful, scary, creepy: Perceptions of online behavioral advertising," in *Eighth Symp. on Usable Privacy and Security*. ACM, 2012, pp. 4:1–4:15.

[2] S. Guha, B. Cheng, and P. Francis, "Challenges in measuring online advertising systems," in *10th ACM SIGCOMM Conf. on Internet Measurement*, 2010, pp. 81–87.

[3] R. Balebako, P. Leon, R. Shay, B. Ur, Y. Wang, and L. Cranor, "Measuring the effectiveness of privacy tools for limiting behavioral advertising," in *Web 2.0 Security and Privacy Wksp.*, 2012.

[4] C. E. Wills and C. Tatar, "Understanding what they do with what they know," in *2012 ACM Wksp. on Privacy in the Electronic Society*, 2012, pp. 13–18.

[5] L. Sweeney, "Discrimination in online ad delivery," *Commun. ACM*, vol. 56, no. 5, pp. 44–54, 2013.

[6] B. Liu, A. Sheth, U. Weinsberg, J. Chandrashekar, and R. Govindan, "AdReveal: Improving transparency into online targeted advertising," in *Twelfth ACM Wksp. on Hot Topics in Networks*. ACM, 2013, pp. 12:1–12:7.

[7] P. Barford, I. Canadi, D. Krushevskaja, Q. Ma, and S. Muthukrishnan, "Adscape: Harvesting and analyzing online display ads," in *23rd Intl. Conf. on World Wide Web*. International World Wide Web Conferences Steering Committee, 2014, pp. 597–608.

[8] M. Lécuyer, G. Ducoffe, F. Lan, A. Papancea, T. Petsios, R. Spahn, A. Chaintreau, and R. Geambasu, "XRay: Increasing the web's transparency with differential correlation," in *USENIX Security Symp.*, 2014.

[9] S. Englehardt, C. Eubank, P. Zimmerman, D. Reisman, and A. Narayanan, "Web privacy measurement: Scientific principles, engineering platform, and new results," Manuscript posted at http://randomwalker.info/publications/WebPrivacyMeasurement.pdf, 2014, accessed Nov. 22, 2014.

[10] A. Sabelfeld and A. C. Myers, "Language-based information-flow security," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 1, pp. 5–19, 2003.

[11] N. R. Wagner, "Fingerprinting," in *1983 IEEE Symp. on Security and Privacy*, 1983, p. 18.

[12] M. Swanson, M. Kobayashi, and A. Tewfik, "Multimedia data-embedding and watermarking technologies," *IEEE*, vol. 86, no. 6, pp. 1064–1087, 1998.

[13] B. Chor, A. Fiat, and M. Naor, "Tracing traitors," in *14th Annual International Cryptology Conf. on Advances in Cryptology*. Springer-Verlag, 1994, pp. 257–270.

[14] Symantec, "Symantec data loss prevention." [Online]. Available: http://www.symantec.com/data-loss-prevention

[15] RSA Labs, "RSA data loss prevention." [Online]. Available: http://www.emc.com/security/rsa-data-loss-prevention.htm

[16] P. Wright, *Spycatcher: The Candid Autobiography of a Senior Intelligence Officer*. Viking Adult, 1987.

[17] L. Spitzner, "Honeytokens: The other honeypot," Symantec Connect Security article, 2010. [Online]. Available: http://www.symantec.com/connect/articles/honeytokens-other-honeypot

[18] P. Papadimitriou and H. Garcia-Molina, "Data leakage detection," *IEEE Trans. on Knowl. and Data Eng.*, vol. 23, no. 1, pp. 51–63, 2011.

[19] M. Monmonier and H. J. de Blij, *How to Lie with Maps*, 2nd ed. University of Chicago Press, 1996.

[20] J. McLean, "Security models and information flow," in *IEEE Computer Society Symp. on Research in Security and Privacy*, 1990, pp. 180–187.

[21] M. Mowbray, "Causal security," in *Computer Security Foundations Wksp.*, 1992, pp. 54–62.

[22] J. A. Goguen and J. Meseguer, "Security policies and security models," in *IEEE Symp. on Security and Privacy*, 1982, pp. 11–20.

[23] J. Pearl, *Causality*. Cambridge University Press, 2009.

[24] P. Good, *Permutation, Parametric and Bootstrap Tests of Hypotheses*. Springer, 2005.

[25] M. C. Tschantz, A. Datta, A. Datta, and J. M. Wing, "A methodology for information flow experiments," ArXiv, Tech. Rep. arXiv:1405.2376v1, 2014.

[26] A. Datta, M. C. Tschantz, and A. Datta, "Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination," in *Proc. on Privacy Enhancing Technologies (PoPETs)*. De Gruyter Open, 2015, pp. 92–112.

[27] J. R. Ruthruff, S. Elbaum, and G. Rothermel, "Experimental program analysis: A new program analysis paradigm," in *2006 Intl. Symp. on Software Testing and Analysis*. ACM, 2006, pp. 49–60.

[28] P. Sewell and J. Vitek, "Secure composition of untrusted code: wrappers and causality types," in *Computer Security Foundations Workshop, 2000. CSFW-13. Proceedings. 13th IEEE*, 2000, pp. 269–284.

[29] I. Gray, J.W., "Probabilistic interference," in *IEEE Symp. on Research in Security and Privacy*, 1990, pp. 170–179.

[30] J. W. Gray, III, "Toward a mathematical foundation for information flow security," in *IEEE Computer Society Symp. on Research in Security and Privacy*, 1991, pp. 21–34.

[31] D. Volpano, C. Irvine, and G. Smith, "A sound type system for secure flow analysis," *J. Comput. Secur.*, vol. 4, no. 2-3, pp. 167–187, 1996.

[32] G. Barthe, P. R. D'Argenio, and T. Rezk, "Secure information flow by self-composition," in *CSFW '04: 17th IEEE Computer Security Foundations Wksp.*, 2004, p. 100.

[33] N. Vachharajani, M. J. Bridges, J. Chang, R. Rangan, G. Ottoni, J. A. Blome, G. A. Reis, M. Vachharajani, and D. I. August, "RIFLE: An architectural framework for user-centric information-flow security," in *37th Annual IEEE/ACM Intl. Symp. on Microarchitecture*, 2004, pp. 243–254.

[34] J. Newsome and D. X. Song, "Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software," in *Network and Distributed System Security Symp.* The Internet Society, 2005.

[35] V. N. Venkatakrishnan, W. Xu, D. C. DuVarney, and R. Sekar, "Provably correct runtime enforcement of non-interference properties," in *8th Intl. Conf. on Information and Communications Security*. Springer-Verlag, 2006, pp. 332–351.

[36] S. McCamant and M. D. Ernst, "A simulation-based proof technique for dynamic information flow," in *2007 Wksp. on Programming Languages and Analysis for Security*. ACM, 2007, pp. 41–46.

[37] A. R. Yumerefendi, B. Mickle, and L. P. Cox, "Tightlip: keeping applications from spilling the beans," in *4th USENIX Conf. on Networked Systems Design and Implementation*, 2007, pp. 12–12.

[38] G. Le Guernic, "Information flow testing: The third path towards confidentiality guarantee," in *Annual Asian Computing Science Conf.*, 2007.

[39] J. Jung, A. Sheth, B. Greenstein, D. Wetherall, G. Maganis, and T. Kohno, "Privacy Oracle: A system for finding application leaks with black box differential testing," in *ACM Conf. on Computer and Communications Security*. ACM, 2008, pp. 279–288.

[40] R. Capizzi, A. Longo, V. N. Venkatakrishnan, and A. P. Sistla, "Preventing information leaks through shadow executions," in *2008 Annual Computer Security Applications Conf.* IEEE Computer Society, 2008, pp. 322–331.

[41] D. Devriese and F. Piessens, "Noninterference through secure multi-execution," in *2010 IEEE Symp. on Security and Privacy*, 2010, pp. 109–124.

[42] F. B. Schneider, "Enforceable security policies," *ACM Trans. Inf. Syst. Secur.*, vol. 3, no. 1, pp. 30–50, 2000.

[43] D. Garg, L. Jia, and A. Datta, "Policy auditing over incomplete logs: theory, implementation and applications," in *18th ACM Conf. on Computer and Communications Security*, 2011, pp. 151–162.

[44] J. McLean, "A general theory of composition for trace sets closed under selective interleaving functions," in *1994 IEEE Symp. on Security and Privacy*, 1994, p. 79.

[45] D. M. Volpano, "Safety versus secrecy," in *6th Intl. Symp. on Static Analysis*. Springer-Verlag, 1999, pp. 303–311.

[46] R. H. Hoyle, Ed., *Handbook of Structural Equation Modeling*. The Guilford Press, 2012.

[47] D. R. Cox and N. Reid, *The Theory of the Design of Experiments*. Chapman & Hall, 2000.

[48] R. A. Fisher, *The Design of Experiments*. Oliver & Boyd, 1935.

[49] D. R. Cox, *Planning of Experiments*. Wiley, 1958.

[50] P. R. Rosenbaum, "Interference between units in randomized experiments," *J. the American Statistical Association*, vol. 102, no. 477, pp. 191–200, 2007.

[51] S. Greenland, "The logic and philosophy of causal inference: A statistical perspective," in *Philosophy of Statistics*. Elsevier, 2011, pp. 813–830.

[52] L. Sweeney, personal communication, 2015.

[53] J. Ludbrook, "Analysis of 2 $\times$ 2 tables of frequencies: Matching test to experimental design," *International J. Epidemiology*, vol. 37, pp. 1430–1435, 2008.

[54] E. L. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*, 3rd ed. Springer, 2005.

[55] Q. Ma, personal communication, 2014.

[56] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *J. the Royal Statistical Society Series B*, vol. 57, p. 289300, 1995.

[57] J. Y. Halpern and K. R. O'Neill, "Secrecy in multiagent systems," *ACM Trans. Inf. Syst. Secur.*, vol. 12, no. 1, pp. 5:1–5:47, 2008.

[58] C. Dwork, F. Mcsherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography Conf.* Springer, 2006, pp. 265–284.