Information Flow Investigations

Michael Carl Tschantz[a] **Anupam Datta**
Jeannette M. Wing[b]

June 26, 2013
CMU-CS-13-118

[a]Now at University of California, Berkeley
[b]Now at Microsoft Research

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

This research was supported by the U.S. Army Research Office grants DAAD19-02-1-0389 and W911NF-09-1-0273 to CyLab, by the National Science Foundation (NSF) grants CCF0424422 and CNS1064688, and by the U.S. Department of Health and Human Services grant HHS 90TR0003/01. 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.

Keywords: Privacy, Formal Methods, Black Box Analysis

Abstract

Information-flow analysis has largely ignored the setting where the analyst has neither control over nor a complete model of the analyzed system. We formalize such limited information flow investigations and study it in three domains: (1) the determination of what third-party web trackers do with the information they collect, (2) the identification of copyright violators, and (3) the detection of insiders leaking data. We use a common framework based on information flow analysis to relate these three problems while pushing beyond traditional information flow analysis. Viewing these seemingly disparate problems in a common framework allows us to identify the assumptions underlying analyses used for these problems and to show where each area could benefit from the other. Our systematic study leads to practical advice for improving work on tracking web trackers, a previously unformalized area.

1 Introduction

Tracking Web Trackers. Concerns about privacy have led to much interest in determining how third-party associates of first-party websites use information they collect about the visitors to the first-party website. Mayer and Mitchell provide a recent presentation of research that tries to determine what information these third-parties collect [1]. Others have attempted to determine what these third-parties do with the information they collect [2, 3, 4].

The researchers involved in these works each propose and use various analyses to determine what information is tracked and how it is used. They primarily design their analyses by intuition and do not formally present or study their analyses. Thus, questions remain:

1. Are the analyses used sound and/or complete?
2. Are they related to more formal prior work?

To answer these questions, we must start with a formal framework that can express the problem and the analyses. In essence, each of these works is conducting an information flow analysis: the researchers want to know when information flows to a third-party and where it goes from there. Thus, the natural starting point for such a formalism is prior research on *information flow analysis* (IFA). However, despite the great deal of research on IFA (see [5] for a survey), we know of no attempt to relate or inform the research tracking web trackers (TWT) with the models or techniques of IFA, even in an informal manner.

We believe this disconnect exists for an important reason: the traditional motivation for IFA, designing secure programs, pushes it away from analyzing third-party systems as done in TWT. Typically, the analyst is seen as verifying that a system under his control protects information sensitive to 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.

In TWT, the analyzed system is the adversary with the analyst aligned with a *data subject* whose information is collected by the system. In this setting, the analyst has no access to the program running the third-party service, little control over its inputs, and a limited view of its behavior. Thus, the analyst does not have the information presupposed by traditional IFAs. To understand the TWT problem as an instance of IFA requires a fresh perspective on IFA.

Piracy Detection. The implicit assumptions underlying much of IFA research also obscures its connection to another area of research: piracy detection. The cryptography community has done much work on preventing *digital copyright piracy*: the use of computers for the copying and distribution of data in violation of copyright restrictions. While some of this work has focused on actually preventing such piracy with digital rights/restrictions management (DRM), much of it has focused on identifying illicit flows of files to make ramifications achievable. Such work has included *watermarking* [6, 7] and *traitor tracing* [8].

In essence, these works are all IFAs. In particular, the analyst, who is aligned with the copyright holder, would like to determine whether a system (typically a personal computer) is enabling an illicit flow of information. However, those working on these problems have not typically discussed them as such since the problem of piracy detection does not fit into the traditional IFA setting. In particular, the analyst has little if any access or control over the analyzed system. Like with TWT problem, the analyst must investigate an uncontrolled black box. Indeed, we find that some of the intuitive approaches used in TWT are related to cryptographic measures used in piracy detection.

Insider Data Leak Detection. Companies handling sensitive data have adopted a variety of methods to discourage the misuse of such data by their employees. In particular, governments are concerned with employees leaking classified documents to reporters or foreign spies. Thus, they employ counterintelligence operations to detect such leaks and determine the identity of the employee leaking the information.

Furthermore, for ethical reasons and to comply with regulations, such as the HIPAA Privacy Rule [9], healthcare providers limit the use of personal health information. Thus, they would benefit from a method of identifying employees who share such information inappropriately, such as by posting it publicly.

While the motivations behind these methods differ from TWT and piracy detection, these methods are also IFAs. In particular, they are attempting to determine which employee is enabling an illicit flow of information.

Goal and Contributions. Our goal is to systematize the information flow problems and analyses common to these areas of research. To do so, we identify and formalize the limited abilities of the analyst in these problems as the setting of *information flow investigations*, a form of analysis between the extremes of white box program analysis and black box monitoring. We show that the ability of the analyst to control some inputs during an investigation enables powerful *sting* analyses that *setup* the system in question to discover its use of information without a white box model of the system. Our investigation framework provides a fresh perspective on our diverse set of motivating applications and allows us to elucidate and challenge approaches in these areas and in IFA.

We start by examining our motivating applications of TWT, piracy detection, and data leak detection. After describing the problems and solutions to them in Section 2, we discuss IFA in general and limitations of traditional IFA in Section 3. We abstract over these problems to show that they are all instances of IFA. In particular, they belong to a class of IFA we have identified as *investigations* that shifts 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. This work systematizes the common but hitherto independent efforts of our motivating applications by formalizing them all as investigations.

In Section 4, we present a formalization of investigations that cleanly describes the common characteristics of these problems. We formalize investigations in terms of a version of noninterference, the primary formal definition of traditional IFA [10]. We present all three applications in terms of noninterference showing the relationship among these areas. Furthermore, we give the first formal characterization of the TWT problem. We discuss the soundness and completeness of detecting (non)interference by investigation in general. We conjecture that sound or complete detection requires additional assumptions, which we discuss in the next section.

We identify, in Section 5, a class of investigations we call *stings* that can produce strong guarantees under reasonable assumptions. They leverage the ability to control some inputs to the system to *setup* the system in such a way that its outputs reveals information use. These analyses resemble the inductive reasoning used in experimental sciences. For each analysis used in the prior work on our motivating applications, we formalize it as a sting, which shows their similarities and differences. We derive the assumptions implicit in the informal analyses used in practice by studying them in our formalism as stings. These assumptions qualify the soundness and completeness of the conclusions drawn by works using stings.

In particular, we provide practical suggestions, which are summarized in Section 6, for conducting future TWT analyses by applying our framework to prior works in the area. In particular, we make suggestions for producing more assured results in the area. We end with directions for future work based on these suggestions and with new directions for research that apply investigations to other security problems outside of IFA.

Systematization of investigations is becoming increasingly important as technology trends (e.g., Cloud and Web services) result in analysts 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 ties together seemingly disparate areas of security and privacy by placing them in the context of information flow investigations.

2 The Motivating Applications

In this section, we discuss various application domains that have been treated separately in the past. For each domain, we discuss specific problems and approaches for solving them. In the following sections, we argue that each of these domains yield problems solved by investigating information flows.

Our goal is not a detailed survey of these research areas. Rather, we provide an overview of the areas looking for broad classes of problems and analyses that highlight the abilities, limitations, and assumptions of analyses in these areas. To do so, we look at representative works in the areas and refer to surveys where available.

2.1 Piracy Detection

Publishers of copyrighted material would like to detect those infringing their copyrights so that they may seek judicial relief. While not typically viewed as an IFA problem, in essence, the copyright holder would like to determine how his copyrighted data flows through a system and who passes it to whom.

For our purposes, copyright infringement breaks into two forms that differ in whether the infringer's identity is clear from the violation itself. The first type of violation that interests us is plagiarism, in which the identity is clear from the act. The second type is file sharing.

Plagiarism. Plagiarism is a publisher passing off another publisher's work as his own. That is, to plagiarize a work, a publisher must not only copy the work of another but also change it to appear to be his own original work. In many cases, such as novels, the detection of plagiarism is trivial since the probability of the plagiarizing publisher independently producing the same novel as another publisher is negligible. However, in reference works, such as maps or phone books, we expect two publishers to produce nearly identical works. In such cases, one must find evidence of copying to prove plagiarism. Thus, this is a problem of IFA.

One approach the publisher can use for this problem is to employ a *copyright trap*: deliberately unusual (typically, false) information inserted into reference works to detect copying. For example, a map might include a *trap street* that is purposely misplaced and/or misnamed [11]. If another publisher mechanically copies the map, the inclusion of the trap street in the copy will indicate the

copying. (Here, the analyst must be aligned with publisher. We discuss detecting the plagiarism of third-party works only in Section 6.)

File-Sharing Detection. File-sharing occurs when a person illegally copies a copyrighted work and shares it with others. In the copyright trap, the detection of plagiarizing publisher is made easier by the plagiarizer marking the copied map with his identity. In the case of file sharing, the sharer typically does not mark the illegal copy with his identity. Thus, upon finding illegal copies, the copyright holder must determine who provided the copies.

One method of making this determination possible is to include a unique *watermark* embedded into each copy of the data released by the legal publisher. (See e.g., [6, 7] for overviews.[1]) The watermark can include the identity or a key that links to the identity of the person to whom the publisher sold the copy. The publisher can construct the watermark and/or mapping either by recording the information at the point of sale as done by iTunes [13] or by using activation codes afterward. Publishers attempt to make watermarks *robust* against removal by interspersing it within the host data or hiding it with steganography, but some techniques such as dithering can probabilistically remove watermarks [7].

A special instance of catching those who share files is *traitor tracing*, which is a method of determining who provided cryptographic keys to enable decrypting copyrighted data [8]. In this case, for efficiency, a single instance of the data is broadcast publicly, but is encrypted to ensure that only key holders can view the data. The keys are distributed individually allowing them to act like a watermark and identify anyone publicly posting his key. This system and watermarking in general must be made secure against reasonably sized coalitions of copyright pirates who can combine their keys or watermarked data to avoid identification.

2.2 Insider Data Leak Detection

While copyright pirates typically have no affiliation with the copyright holders, the problem of data leak detection deals with the insider threat as part of data loss prevention [14, 15]. For example, upon learning that classified information is being leaked to the press, an investigative agency needs to determine the identity of the information leaker. As with the file sharer in copyright piracy, the leaker typically leaks the information anonymously.

In such cases, investigators have employed *Barium meals*, a watermarking-like analysis [16]. To use a Barium meal, the investigators feed different versions of classified information to each suspect leaker. While the investigators cannot see what each suspect does with this information directly, they may be able to infer the identity of the leaker based upon newspaper accounts of the leaked information.

Similar techniques have been employed in other investigations of insiders. For example, the former technology company Orbious automated the process of running a Barium meal for wide-scale use in the corporate setting [17]. A company can distribute email lists to business partners with varying fake addresses, or *honeytokens* [18]. The company releasing the list monitors the fake email accounts to identify any partners misusing the list. Papadimitriou and Garcia-Molina

[1] Wagner [6] considers watermarking to be a type of *fingerprinting* and refers to it as such. Others treat fingerprinting and watermarking as separate methods with fingerprinting being passive (e.g., [12]). Under this more narrow usage, many of Wagner's fingerprinting schemes, those not based on "recognition", would be considered watermarking and not fingerprinting.

consider the case where fabricated data is unacceptable but varying subsets of a database can be released to suspicious partners [19].

Each of these methods bears similarities to watermarking in that variations of a sensitive document are released to detect information leaks.

2.3 Tracking Web Trackers

Many first-party websites that users intentionally visit use services from third-party companies for features such as ad placement and social media. To use these services, the first-party websites provide the third-party services with information about their visitors. The third-parties often span numerous websites and attempt to track the visitor across the web by associating a single identity with each user.

A visitor might be unaware of the third-parties and surprised by the amount of information that they can aggregate about him. Thus, such third-party tracking is concerning to consumer advocates, regulatory bodies, and privacy researchers. (For a survey, see [1].)

Researchers, in particular, are attempting to track the web trackers (TWT) by determining the information third-parties collect and how they use that information. These "fourth-party" [1] researchers often pose as normal website visitors and study the information sent to the third-parties and the advertisements they receive back from them. However, the researcher's interactions with the third-parties are limited requiring the researchers to draw conclusions from limited information.

In essence, the fourth-party researchers are attempting to determine whether certain information flows to the third-parties and if so, how it is used. Thus, they are performing information flow analysis.

The fourth-party researchers have invented intuitive but ad hoc measures. In the simple case, the researcher would just like to determine what information a first-party website causes a browser to send to a third-party. Such analyses can use a proxy to detect information sent to the third-parties. For example, Krishnamurthy et al. conducted a study to determine that the majority of examined websites provide some private information to third-parties [20]. They make this determination by using the Fiddler web proxy to watch the information that their browser conveys to third-parties when interacting with a first-party website. For example, they find that the referrer field in HTTP GET requests resulting from a first-party website sometimes includes the visitor's email address since it was part of the first-party URI.

Other studies would like to determine what the third-parties do or can infer from the information that is provided to it. For example, Wills and Tatar studied how Google selects ads based on information provided by the website visitor via first-party websites [3]. The authors draw conclusions about Google's information use in two ways. First, they observed Google showing them (posing as normal website visitors) ads that included sensitive information they provided to Google by interacting with a website that uses a Google service, such as Ad Sense. Second, when posing as two different users with different interests, they observed Google showing them ads differing in ways related to the differing interests.

Guha et al. study a similar problem using a more statistical approach [2]. Like Wills and Tatar, they would pose as various visitors with different characteristics. To test whether some change between two user profiles resulted in a change in Google's ads, they would pose as the first profile twice and as the second profile once. By using the same profile twice, they could calculate the baseline amount of noise or "ad churn" in the ads independent of the change. If the change between the first and second profile is larger than this baseline, they then conclude that the change

between profiles caused the increased difference in the ads. Balebako et al. adopt the methodology of Guha et al. to study the effectiveness of web privacy tools [4]

While Wills and Tatar look at the differences between ads to determine whether they have anything to do with sensitive information, Guha et al. do not attempt to interpret the ads to see what could have caused the change. (They did look at the ads while validating their analysis.) While the analysis of Wills and Tatar leads to a better understanding of how the website is using the information, the analysis of Guha et al. can find changes that people are apt to miss since the relationship between the changes in input and output are not immediately clear or because they take a larger sampling to notice than is possible with manual inspection. For example, they find that a profile purportedly of a homosexual male gets a large increase in nursing school ads, which may have been missed by the Wills and Tatar's method since there is no clear connection between the change in the profile to the change in the ads. As Guha et al. point out, this lack of connection makes this discovery more important since the website visitor would also be unlikely to realize that responding to the nursing ad could leak sensitive information to the nursing program.

3 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 end by discussing the capabilities of the analyst in our motivating applications and how these analyses are inappropriate given these capabilities.

3.1 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 [10]. Intuitively, noninterference requires that the system behaves identically from the perspective of untrusted users regardless of any sensitive inputs to the system.

Noninterference involves a system with various I/O channels. One is called H and represents high-level information. Another channel is called L and represents low-level information. 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 channel H from reaching channel L.

To formalize this goal, we suppose that the system in question Q sends and receives (possibly over multiple channels) messages from the set M. From the set of messages M, we select two disjoint subsets H and L. Typically, H corresponds to all messages to and from high-level users, and L to all messages to or from low-level users. However, we often have a single user sending messages from both sets and do not require H and L to partition M, creating a primitive form of intransitive noninterference [21, 22]. For a sequence $\vec{m}$ in M^*, let $\lfloor \vec{m}{\downarrow}L \rfloor$ represent $\vec{m}$ restricted to only those messages that are in L. That is, it "purges" all high-level messages. Formally, for a

subset M' of M,

$$\lfloor m{\cdot}\vec{m}{\downarrow}M' \rfloor = \begin{cases} m{\cdot}\lfloor \vec{m}{\downarrow}M' \rfloor & \text{if } m \in M' \\ \lfloor \vec{m}{\downarrow}M' \rfloor & \text{otherwise} \end{cases}$$
$$\lfloor [\,]{\downarrow}M' \rfloor = [\,]$$

where $[\,]$ is the empty sequence, and $m{\cdot}\vec{m}$ is the sequence $\vec{m}$ with m prepended to it. (We will abuse notation and use $\cdot$ for appending and sequence concatenation as well.) Similarly, let $\lceil \vec{m}{\uparrow}H \rceil$ be $\vec{m}$ with all the messages in H removed.

The set M is also partitioned into two sets I_q and O_q with I_q being the set of messages that are inputs to Q and O_q being the set of messages that are outputs from Q. Given a sequence of inputs $\vec{\imath}$ in I_q^*, we let $Q(\vec{\imath})$ represent the result of the system Q running on inputs $\vec{\imath}$.

Definition 1. *A system Q has* noninterference *from L to H of M iff for all input sequences $\vec{\imath}_1$ and $\vec{\imath}_2$ in I_q^*,*

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

Intuitively, this definition says that if inputs only differ in high-level messages, then the same low-level outputs should result from the system with complete disregard for what if any high-level inputs were provided to the system.

3.2 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. Proving (non)interference by brute force is difficult for systems with many possible inputs especially when the system, its inputs, or its outputs are out of the control or view of the analyst. Thus, analysts must employ strategic analyses specialized to his capabilities.

IFA grew out of the demand to build military computers respecting mandatory access controls (MAC). Thus, much of the work in the area presumes that the analyst has a degree of control over the production of the analyzed system. In the setting most favorable to the analyst, he can have the program written in a special-purpose programming language that has features such as type checking for detecting information flows [23]. (See [5] for a survey.) Less fortunate analysts may have to analyze a program written in a standard programming language. A static analyses useful in this setting is a modified form of model checking [24]. A dynamic approach can run the program after instrumenting the code to track values carrying high-level information (e.g., [25, 26, 27, 28]).

The above methods are inappropriate for TWT 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 exist that operate by running the program multiple times with varying inputs to detect changes in output that imply interference [29, 30, 31]. 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 [29]. In theory, such black box analyses could be modified to not require any access to code by completely

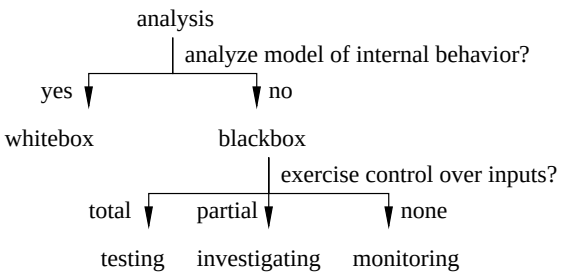

Figure 1: Taxonomy of analyses

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 and testing notions of equivalence (e.g., [32, 33]).

Testing will not work for our applications. For example, in the application of TWT, the analyst cannot 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*. Monitoring passively observes the execution of a system. For example, Schneider's *execution monitoring* does not presume complete control over the system's environment and does not provide any inputs to the system (other than to terminate it in cases of violation) [34]. Furthermore, it neither uses a model of the system nor modifies the system. While some monitors are too powerful by being able to observe the internal state of the running system (allowed in [34]), others match our needs in that the analyst only has access to a subset of the program's outputs (e.g., [35]). However, all monitors are too weak since they cannot provide inputs to the system as our application analysts do. We need a form of black box analysis between the extremes of testing and monitoring.

Thus, we find that no prior work on IFA that corresponds to the capabilities of the analyst in TWT. In the next section, we will formalize these capabilities as the ability to conduct an *investigation*. Investigation may be viewed as an interactive extension of a limited form of execution monitoring that allows inputs from analyst but limits the analyst to only observing a subset of system I/O.

3.3 Investigating Information Flows

Unlike the primary motivation of traditional IFA, developing programs with 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

- has no model of or access to the program running the system,
- cannot observe the internal states of the system,

- has limited control over and knowledge of the environment of the system,
- can observe a subset of the system's outputs, and
- has control over a subset of the inputs to the system.

We will call performing IFA in this setting *investigating*. Investigation may be viewed as an interactive extension of a limited form of execution monitoring that allows for analyst inputs but limits the analyst to only observing a subset of system I/O.

Investigating information flows produce more qualified results than static analyses. However, we show that interesting results are possible. In particular, we will explore what additional assumptions about the system in question will imply the soundness or completeness of various analyses in the settings of our motivating applications. These settings vary in the analyst's abilities. Thus, we start by re-formalizing IFA with an explicit analyst.

4 Formal Models

In this section, we formally model the setting of investigation in general and show that each of our motivating applications fits into our model. In the next section, we model the individual investigative techniques found in our applications.

We are concerned with the entire setting of investigation since we would like to prove general results about when an analyst can investigate successfully. Thus, to allow us to parametrize our results over individual investigative techniques, we need a formalism that explicitly models the analyst (e.g., a fourth-party researcher) in addition to the system in question (e.g., Google). We present such a model, relate it to the traditional definition of noninterference, and use it to discuss the possibility of investigations being sound or complete in general.

However, to be concrete, we will first present our model specialized for TWT. Later in this section, we discuss generalizations of this model and using it for other applications.

4.1 Modeling Networks of Systems

In TWT, a fourth-party researcher acts as an analyst a of a system in question q, such as Google's AdSense. The system in question q runs a process Q while the analyst a runs an analysis A (also a process). We denote these facts by $q{:}Q$ and $a{:}A$, respectively. While the analyst knows the identity of the system (i.e., q), it does not know a priori its process Q. The analyst a uses the analysis A in an attempt to determine whether the process Q uses some sensitive information that it receives from first-party websites $f_1, \ldots, f_n$ using channels that a cannot observe [2, 3, 4]. Complicating matters, a does not know the processes $F_1, \ldots, F_n$ ran by the first parties. Furthermore, these systems operate in an environment containing other systems (e.g., other website visitors) that might provide confounding inputs to any of them. We use $\vec{e}{:}\vec{E}$ to denote the environment containing various other systems whose processes $\vec{E}$ might be unknown to the analyst a.

Using $||$ to denote the parallel composition of systems into a network, the network created by all of these systems is $a{:}A||q{:}Q||f_1{:}F_1|| \cdots ||f_n{:}F_n||\vec{e}{:}\vec{E}$. The systems each have a channel to every other system to communicate with one another by passing messages. We assume that the identity of sender of each message is apparent to the receiver, but systems may send anonymous messages by sending them via an intermediate system. We use O_s to denote the messages that system s can send as output, I_s to denote those s can receive or accept as inputs, and M_s to denote $O_s \cup I_s$.

We use $[a{:}A||q{:}Q||\vec{e}{:}\vec{E}]$ to denote the messages sent between all pairs of processes in the network in temporal order. (For brevity, we have pushed $f_1{:}F_1||\cdots||f_n{:}F_n$ into the environment $\vec{e}{:}\vec{E}$.) In general, this is a mapping from time to every message sent at that time. For simplicity, we typically abstract to a sequence of messages by assuming that no two messages are sent at the same time and ignoring the time between each message.

We assume this sequence is uniquely defined by assuming that the network is a closed deterministic system. Furthermore, we assume that each of the systems is deterministic given the inputs to it. These assumptions are reasonable since every system in which we are interested becomes deterministic and non-probabilistic given all the inputs to it. In essence, modeling enough inputs to a system moves nondeterminism and probabilistic behavior from within the system to uncertainty about the system's environment $\vec{e}{:}\vec{E}$. For example, difficult to predict behavior typically modeled as coming from a truly nondeterministic random number generator is actually from uncertainty about a seed provided to a pseudo-random number generator. Thus, we can represent a system using such behavior as a deterministic system accepting a seed as an input from some process E in the environment $\vec{e}{:}\vec{E}$ where the process E, and, thus, the seed it provides, is unknown to the analyst. Similarly, the environment could include any schedulers used to determine the order in which systems execute in parallel, thereby, converting nondeterminism from scheduling itself into uncertainty over the process that schedules. As another example, while for simplicity the order in which packets arrive from a network is typically modeled as nondeterministic, since we can use the environment $\vec{e}{:}\vec{E}$ to quantify over the entire network, this order can be a deterministic function of it. (For a related algorithm, see [36].)

The result of our assumptions is that each system and network, as a whole, become deterministic. (For further discussion, see [37].) Furthermore, while the systems in $\vec{e}{:}\vec{E}$ could be very difficult to model for a white box analysis, since we are focused on black box analysis, we have no need to model them beyond their interfaces with a and q. The assumption of determinism is important since while there are many competing generalizations of noninterference to the nondeterministic setting (e.g., deducibility security [38], generalized noninterference [39], restrictiveness [39], nondeducibility on strategies [40], generalized noninference [41], separability [41], and prefect security property [42]), the main competitors collapse into standard noninterference in the deterministic case [43].

To define $Q(\vec{\imath})$ in terms of our model, we use general compositions of the form $q{:}Q||\vec{e}{:}\vec{E}$ where $\vec{e}{:}\vec{E}$ includes all systems other than q, including $a{:}A$. Let $\mathbf{eE}_{\vec{\imath}}$ be the set that contains all environments $\vec{e}{:}\vec{E}$ that produce $\vec{\imath}$ as q's inputs, i.e., $\vec{e}{:}\vec{E}$ such that $\lfloor[q{:}Q||\vec{e}{:}\vec{E}]{\downarrow}I_q\rfloor = \vec{\imath}$. Since $q{:}Q$ is deterministic given its input, for any $\vec{e}_1{:}\vec{E}_1$ and $\vec{e}_2{:}\vec{E}_2$ in $\mathbf{eE}_{\vec{\imath}}$, $\lfloor\vec{m}_1{\downarrow}M_q\rfloor = \lfloor\vec{m}_2{\downarrow}M_q\rfloor$ where $\vec{m}_1 = [q{:}Q||\vec{e}_1{:}\vec{E}_1]$ and $\vec{m}_2 = [q{:}Q||\vec{e}_2{:}\vec{E}_2]$. This common sequence of messages $\lfloor\vec{m}_1{\downarrow}M_q\rfloor = \lfloor\vec{m}_2{\downarrow}M_q\rfloor$ is $Q(\vec{\imath})$.

4.2 Information Flow Investigations

For TWT, we are interested in the ability of the analyst a to conduct a sound or complete analysis for determining whether a web tracker in question q uses some sensitive information for purposes such as advertising. We formalize this determination as determining whether q has interference from the sensitive information to outputs such as ads.

Recall that interference for q runs from a high-level set H of q's inputs to a low-level set L of q's outputs. Typically in TWT, L is the set of ads that q can show a on websites. The high-level set H is the set of messages containing sensitive information to q from a or a first-party website f_j. The sets H and L need not partition the set M_q of q's messages since some messages might neither

contain sensitive information nor represent a concerning use of sensitive information. (Since H and L need not partition M_q, we will have a primitive form of intransitive noninterference [21, 22].)

For a network $\mathcal{N} = a{:}A||q{:}Q||\vec{e}{:}\vec{E}$, we say that the analyst system $a{:}A$ sends a *positive result* if it sends a distinguished message `1` on some distinguished channel. We say that a is *sound* for interference of q in $\mathcal{N}$ iff it returning a positive result implies that the process Q has interference. We say that $a{:}A$ is *complete* for interference of q in $\mathcal{N}$ iff Q having interference implies that $a{:}A$ will return a positive result. If for all processes Q and $\vec{E}$, $a{:}A$ is a sound (complete, resp.) analysis for interference of $q{:}Q$ in $a{:}A||q{:}Q||\vec{e}{:}\vec{E}$, then we say that a is a sound (complete, resp.) analysis for interference of q.

The above notion of analyst captures the idea of investigation since the analyst a cannot directly examine Q and can only learn of Q by interacting with it in an unknown environment. For $a{:}A$ to be a sound or complete analysis, it must produce correct results for all such Q and all $\vec{E}$ without knowing a priori the value of Q. If we allow A to vary with Q, we would get a white box analysis. Allowing A to select the environment $\vec{e}{:}\vec{E}$ and observe all messages results in testing.

This setup shows how TWT differs from traditional IFAs in the security setting. In particular, the analyst a can be both the high-level source of sensitive information from H and the low-level sink observing ads from L. In the traditional setting, the source and sink are different allowing one to conclude that noninterference provides the source with confidentiality from the sink. Here, the sink already knows the actions of the source since they are both the analyst. However, the analyst does not know how the process Q uses inputs from a to it. Thus, it is Q, and not the high-level inputs, that has an element of secrecy about it. By being an operational property about how inputs affect outputs rather than an epistemic property (cf. [37]), noninterference can formalize this non-traditional IFA.

4.3 Limitations of Information Flow Investigations

Prior work shows that no monitor can detect information flows [41, 34, 44]. We argue that investigation, with the additional ability to control some inputs to the system, does not improve upon this situation. In particular, we argue that no non-degenerate analysis can be either sound or complete for interference. (Our usage "sound" and "complete" is the reverse of that in work on static analysis since we focus on detecting interference rather than verifying noninterference.)

For the unsoundness argument, we consider an arbitrary process $q{:}Q$ for which A results a positive result indicating interference. Since $a{:}A$ is an investigation, its decision is based solely upon its interactions with other systems. Thus, it will return the same positive result for a process Q_{N} that always produces the same outputs as Q did irrespective of its inputs. Since Q_{N} always produces these outputs, we would expect it to have noninterference making A's positive result false.

The argument for incompleteness is symmetric. Our arguments implicitly assume that the internal structure of systems is rich enough to construct process like Q_{N}. Exploring what internal structures enables or disables such a processes would take us too far afield from our goals of black box analysis. Thus, we leave formalizing these arguments to future work.

Conjecture 1 (Unsoundness)**.** *For any analysis $a{:}A$ and non-empty H and L, if A ever returns a positive result, then A is unsound for interference from H to L.*

Conjecture 2 (Incompleteness)**.** *For any analysis $a{:}A$ and non-empty H and L, if A ever returns a negative result, then A is incomplete for interference from H to L.*

4.4 Other Problems

Here, we show that our formalism of a network of interacting systems can model problems from our other application domains.

4.4.1 Piracy Detection: Plagiarism Detection

Consider a publisher who would like to detect whether a rival publisher ever plagiarizes its publications. Since reference works such as phone books and maps should not vary across publishers, the similarity of these works do not necessarily imply plagiarism. The publisher must show that the rival actually copied its work. That is, it must illustrate a flow of information from its own publications to the rival's.

To formally model this problem, let the publisher be the analyst $a{:}A$ and rival publisher be the system in question $q{:}Q$. Let Pubs_a be the set of messages containing a publication of a and let Pubs_q be q's publications. Since publications are public, they are sent in messages to every other system. We assume the publications identify their publisher, but can be otherwise identical.

To see that this problem is an IFA, let the set H of high-level inputs to q be Pubs_a and the set L of low-level outputs from q be Pubs_q. Since for the purposes of this analysis, the analyst does not care where its publications flow other than to q's publications, no other outputs belong in L. Since the analyst does not care whether q uses other information, the analyst must keep everything other than the Pubs_a fixed across comparisons. Thus, no other inputs should be in H.

The network is $a{:}A||q{:}Q||\vec{e}{:}\vec{E}$. The goal of the analyst a is to be a sound and complete analysis for interference by q from H to L. However, the analyst only knows the value of A and not of Q or $\vec{E}$, which could present confounding inputs to q. Thus, the analyst can use neither white box analysis nor testing for this problem. However, in addition to being able to observe the outputs of q in L, a has control over the inputs to q in H. Thus, the analyst may go beyond monitoring to investigating.

4.4.2 Data Leak Detection: Leaking Classified Documents

In plagiarism detection, the rival made any copied documents publicly available in an attributable form to others including the analyst. In the case of an insider leaking classified documents, the insider may do so in a clandestine manner, which complicates the problem. In particular, the insider may leak the document to an outsider, such as a reporter, who makes the document public without revealing its source. An analyst attempting to determine the identity of the leaking insider must do without observing the leaker providing the sensitive document to the publishing outsider.

More formally, the analyst a would like to determine whether an insider q relayed information provided by a to another system p that publishes information. Complicating matters, a cannot directly observe the communications between q and p and q is typically just one of a set of possible sources of information to p.

To express this problem as an investigation, let H be the set of classified messages a can send to q. Let L be the set of messages that q can send to p. The analyst a receives messages from Pubs, the set of p's publications.

The network is $a{:}A||q{:}Q||p{:}P||\vec{e}{:}\vec{E}$. The goal of the analyst a is to determine whether q has interference from H to L. However, the analysis only knows the value of A and not of Q, P, or $\vec{E}$. Furthermore, a does not get to directly observe the outputs of q in L, which are directed to p instead. However a can control the inputs to q in H. Thus, this problem is also an investigation.

4.4.3 Abstractions over Problems

In each of the above problems, the analyst could control some of the inputs in H to the system in question q but did not know the process Q ran by q. They differed on whether the analyst could directly observe low-level outputs in L or not. Each of the problems listed in Section 2 is an instance of one of these two settings, or abstract problems. Instances of the setting in which the analyst can directly observe the outputs in L include plagiarism detection and TWT. Problems in which the analyst cannot include file-sharing detection and data leaker identification.

In the next section, we will consider analyses for these forms investigations. We present abstract analyses that aim at all concrete problems of a setting. However, each analysis carries with it a set of assumptions that may be valid for only some instances of a particular setting.

5 Formalization of Stings

For each of our motivating applications, we have already informally presented a variety of analyses for investigation, such as watermarking and fourth-party tracking. In this section, we formalize these analyses in our model.

All four analyses we consider follow a similar pattern whereby before looking for evidence of interference, the analyst first sets up the system in question with inputs designed to make detecting interference easier. Thus, these investigations go beyond *interrogation*, that is, asking questions after a crime (interference) may have been committed to learn about it. Rather, these analyses are each similar to a *sting operation* in which the investigators actively participate in the crime to *setup* the criminal [45]. Such setups allow the analyst to use its ability to control certain inputs to the system and go beyond what is possible with monitoring. The analyst employing these setups resembles a scientist manipulating factors during an experiment.

In terms of the setups used, these stings break into two camps. The *broadcast nonce sting* and the *bilateral nonce sting* both supply a nonce and check for its presence in other messages. The *differencing sting* and *baseline sting* both compare the behavior of the system under multiple inputs. We examine one analysis from each camp in detail; we briefly discuss the other two and hybrids of them.

For each sting, we explore what assumptions will enable us to prove that the sting is sound or complete for interference. Rather than finding the weakest of such assumptions, we will instead focus on intuitive assumptions. These assumptions highlight possible sources of false positives and false negatives. Furthermore, qualified soundness and completeness theorems show that the analyst can safely focus on only these possible sources and the assumptions made while modeling the study.

For each sting, we also examine its use in actual TWT studies [3, 2]. We describe how their use either conforms to or deviates from the assumptions needed for soundness or completeness. We select and scrutinize these studies because they contain interesting and important results that we would like to place into the context of IFA; not because we believe them to contain major flaws. Our formalism is an abstraction of the actual problems facing researchers and does not, for example, include the statistical inferences that drive empirical science. Nevertheless, we both elucidate aspects of these studies and exercise our formalism.

Each of these stings is an abstraction of one or more applied stings used in practice. Table 1 summarizes these relations.

Table 1: Stings: settings (columns), approaches (rows), abstract (boldface), and applied (standard face)

	Direct Observation Needed	Direct Observation Not Needed
Nonce Based	**Broadcast Nonce**: TWT [3], copyright trap [11]	**Bilateral nonce**: Barium meal [16], watermarking [7], traitor tracing [46]
Comparison Based	**Differencing**: TWT [3], traitor tracing [46] **Baseline**: TWT [2]	(area of future research)

5.1 Differencing Sting

Consider the TWT study of Wills and Tatar in which they pose as various visitors to first-party websites [3]. Intuitively, they do so to run Google's ad service multiple times on varying inputs. We generalize and formalize this reasoning as the *differencing sting*. By simulating the comparison of two runs of a system, this sting is the most directly related to the definition of noninterference. Despite this close relation, the assumptions that prove soundness or completeness are complex, showing the importance of the careful reasoning made possible by our formalism.

The differencing sting consists of the analyst $a{:}A$ (e.g., Wills and Tatar) providing the analyzed system $q{:}Q$ (e.g., Google) a sequence of inputs $\vec{\imath}^a$ that breaks into two parts $\vec{\imath}^a_1$ and $\vec{\imath}^a_2$. Intuitively, each part simulates two different runs that do not differ in low-level inputs: $\lfloor \vec{\imath}^a_1 {\downarrow} L \rfloor = \lfloor \vec{\imath}^a_2 {\downarrow} L \rfloor$. These input sequences are interleaved with confounding inputs to q from other systems in the environment (e.g., other users of Google) to create the input sequences $\vec{\imath}_1$ and $\vec{\imath}_2$ on which q runs. The analyst then compares two *sub-runs* of the system: the behaviors of q under $\vec{\imath}_1$ to the behavior under $\vec{\imath}_2$.

While studies often interleave the inputs of $\vec{\imath}^a_1$ and $\vec{\imath}^a_2$ over time, for simplicity, we will assume the analyst runs one sub-run after the other (i.e., $\vec{\imath}^a = \vec{\imath}^a_1 {\cdot} \vec{\imath}^a_2$). We will also assume that q produces all its outputs caused by $\vec{\imath}^a_1$ before receiving any inputs from $\vec{\imath}^a_2$. For example, the analyst can log into one profile, observe interactions with Google, log out of that profile and into another, and then observe interactions under the second profile.

Ideally, these two sub-runs would be equal to the two runs that system q would produce if the analyst could run it twice, once on $\vec{\imath}^a_1$ and once $\vec{\imath}^a_2$. However, the two sub-runs produced by $\vec{\imath}^a_1$ and $\vec{\imath}^a_2$ are not actually two runs of the system. Since the analyst cannot restart the system, the two sub-runs might not start from equivalent initial states. Furthermore, differences can arise from confounding inputs from systems in the environment that are not under a's control. Soundness requires that these confounding inputs do not cause q to behave differently on $\vec{\imath}^a_2$ than it would had had $\vec{\imath}^a_2$ been q's only inputs from a.

To formalize this requirement, we must formalize the sequences $\vec{\imath}_1$ and $\vec{\imath}_2$ that arise from the uncontrolled environment interleaving confounding inputs with the sequences $\vec{\imath}^a_1$ and $\vec{\imath}^a_2$, respectively. To formalize $\vec{\imath}_1$, let $\mathcal{U}_1(\vec{m}, \vec{\imath}^a_1)$ denote all the inputs to q in $\vec{m}$ before the first input of $\vec{\imath}^a_2$. Formally, $\mathcal{U}_1(\vec{m}, \vec{\imath}^a_1)$ is the longest prefix $\vec{\imath}_1$ of $\vec{\imath} = \lfloor \vec{m} {\downarrow} I_q \rfloor$ such that $\lfloor \vec{\imath}_1 {\downarrow} M_a \rfloor = \vec{\imath}^a_1$. For $\vec{\imath}_2$, let use $\mathcal{U}_2(\vec{m}, \vec{\imath}^a_1)$ denote the remaining inputs to q after $\vec{\imath}_1$. That is, $\mathcal{U}_2(\vec{m}, \vec{\imath}^a_1)$ is the input sequence $\vec{\imath}_2$ such that $\lfloor \vec{m} {\downarrow} I_q \rfloor = \vec{\imath} = \vec{\imath}_1 {\cdot} \vec{\imath}_2$ where $\vec{\imath}_1 = \mathcal{U}_1(\vec{m}, \vec{\imath}^a_1)$.

These definitions enable defining a homomorphism-like property over the concatenation of inputs:

Definition 2 (Independent Sub-Runs)**.** *A system $q{:}Q$ has* independent L sub-runs for $a{:}A$ providing $\vec{\imath}^a = \vec{\imath}^a_1 {\cdot} \vec{\imath}^a_2$ *iff for all environments $\vec{e}{:}\vec{E}$, $\lfloor Q(\vec{\imath}_1 {\cdot} \vec{\imath}_2) {\downarrow} L \cap M_a \rfloor = \lfloor Q(\vec{\imath}_1) {\downarrow} L \cap M_a \rfloor {\cdot} \lfloor Q(\vec{\imath}_2) {\downarrow} L \cap M_a \rfloor$ where $\vec{m} = [a{:}A || q{:}Q || \vec{e}{:}\vec{E}]$, $\vec{\imath}_1 = \mathcal{U}_1(\vec{m}, \vec{\imath}^a_1)$, and $\vec{\imath}_2 = \mathcal{U}_2(\vec{m}, \vec{\imath}^a_1)$.*

A system q having independent sub-runs is only useful to the analyst if the analyst knows where the first sub-run ends and the second begins. Thus, this sting also requires a procedure that splits the outputs of q into either the sub-run of $\vec{\imath}_1$ or of $\vec{\imath}_2$. In practice, this determination is made simple by the analyst providing q some distinguished input, such as a request to log out of a website. The logout screen identifies the last output ascribed to $Q(\vec{\imath}_1)$.

Formally, we model a's method of splitting outputs of q into sub-runs with a function s_1 that identifies the messages ascribed to $Q(\vec{\imath}_1)$. If $\vec{m}$ is the sequence of all the messages of the network, s_1 operates on $\vec{m}^a = \lfloor \vec{m} \downarrow M_a \rfloor$, those messages visible to the analyst. The result of $s_1(\vec{m}^a)$ is the prefix of $\lfloor \vec{m}^a \downarrow M_q \rfloor$ that the analyst ascribes to $Q(\vec{\imath}_1)$. The analyst ascribes the remainder of $\lfloor \vec{m}^a \downarrow M_q \rfloor$ to $Q(\vec{\imath}_2)$, which we denote as $s_2(\vec{m}^a)$. We use s for s_1 and s_2 collectively.

To characterize s_1 and s_2 correctly splitting the inputs received by a into the two sub-runs, we need to only focus on s_1 since it determines the behavior of s_2.

Definition 3 (Correct Splitting)**.** *For a system q:Q with* independent L sub-runs for a providing $\vec{\imath}^a = \vec{\imath}^a_1 \cdot \vec{\imath}^a_2$*, the splitting function* s_1 correctly L splits *iff* $\lfloor s_1(\vec{m}^a) \downarrow L \rfloor = \lfloor Q(\vec{\imath}_1) \downarrow L \cap M_a \rfloor$ *where* $\vec{m} = [a{:}A || q{:}Q || \vec{e}{:}\vec{E}]$*,* $\vec{m}^a = \lfloor \vec{m} \downarrow M_a \rfloor$*, and* $\vec{\imath}_1 = \mathcal{U}_1(\vec{m}, \vec{\imath}^a_1)$*.*

In summary, if a runs a differencing sting using $\vec{\imath}^a = \vec{\imath}^a_1 \cdot \vec{\imath}^a_2$ and s, then a first provides $\vec{\imath}^a$ to q. Second, it breaks the messages $\vec{m}^a$ that it can observe into $s_1(\vec{m}^a)$ and $s_2(\vec{m}^a)$. Finally, if $\lfloor s_1(\vec{m}^a) \downarrow L \rfloor$ and $\lfloor s_2(\vec{m}^a) \downarrow L \rfloor$ differ, then a returns a positive result, suggesting interference by q.

Soundness. We could prove the soundness of the differencing sting by assuming independent sub-runs, correct splitting, and an additional assumption that $\lceil \vec{\imath}_1 \uparrow H \rceil = \lceil \vec{\imath}_2 \uparrow H \rceil$. This last assumption implies that only the high-level inputs to q vary across the two sub-runs. It allows the analyst to identify changes in the high-level inputs as the cause of any changes in the low-level output.

However, the assumption that $\lceil \vec{\imath}_1 \uparrow H \rceil = \lceil \vec{\imath}_2 \uparrow H \rceil$ holds is unreasonable since a often does not have control over many of the non-sensitive inputs in $\vec{\imath}_1$ and $\vec{\imath}_2$. For example, Google has many confounding inputs not under the control of or even visible to the analyst.

Even if a did control all these confounding inputs, a may have to vary some inputs between $\vec{\imath}^a_1$ and $\vec{\imath}^a_2$ to ensure independent L sub-runs. For example, a might need to use two different user names in the two sub-runs to keep them independent, introducing a confounding input that differs across sub-runs. However, while we expect information associated with a user to affect Google's behavior, we do not expect the choice of user name itself to do so. That is, we would expect the name itself to be noninterfering with the advertisements shown.

This intuition motivates identifying, in addition to H and L, a third set C of message that contains confounding inputs to q that vary across $\vec{\imath}_1$ and $\vec{\imath}_2$ but do not interfere with L. The analyst selects L such that C does not interfere with L and he can control inputs in L: $L \cap I_q \subseteq O_a$. (For simplicity, we assume messages are at a granularity that allows us to identify such confounding information with messages. In practice, the analyst may need to sub-divide messages or use an equivalence relation over them.)

More formally, we replace the assumption that $\lceil \vec{\imath}_1 \uparrow H \rceil = \lceil \vec{\imath}_2 \uparrow H \rceil$ with one that C does not interfere with L and one that $\lfloor \vec{\imath}^a_1 \downarrow L \rfloor = \lfloor \vec{\imath}^a_2 \downarrow L \rfloor$. Under these assumptions, if a sees interference from $H \cup C$ to L, a knows it must have come from H. As a lemma, we prove that if $\vec{\imath}_1$ and $\vec{\imath}_2$ show interference and differ by messages from both H and C but are the same for L, then there exists a third input sequence $\vec{\imath}_3$ such that $\lceil \vec{\imath}_3 \uparrow H \rceil = \lceil \vec{\imath}_1 \uparrow H \rceil$ and it shows interference when compared to q's behavior on $\vec{\imath}_1$.

Theorem 1 (Qualified Soundness)**.** *For all analysts a:A, systems q:Q, environments $\vec{e}$:$\vec{E}$, and partitions of M into L, C, and H such that A is a differencing sting for H to L using $\vec{\imath}^a = \vec{\imath}_1^a \cdot \vec{\imath}_2^a$ and s to split, if the system q has independent L sub-runs for a providing $\vec{\imath}^a$, s_1 correctly L splits, q has noninterference from C to L, $L \cap I_q \subseteq O_a$, and $\lfloor \vec{\imath}_1^a \downarrow L \rfloor = \lfloor \vec{\imath}_2^a \downarrow L \rfloor$, then a positive result from a implies that q has interference from H to L.*

For Wills and Tatar's study, the sensitive information H corresponds to various interests that they induce by visiting first-party websites or using search terms. The low-level messages L correspond to ads. The confounding messages C correspond to inputs from other sources. They attempt to provide independent L sub-runs and correct L splitting by using separate browsers for each sub-run. These assumptions and that C does not interfere with L implies that the actions of one Google user does not affect the ads seen by another.

These assumptions only approximately hold. For example, Google's ads are partly determined by bids made by advertisers, inputs over which a has no control but are likely to affect the ads shown to him. These confounding inputs result in "ad churn" [2].

To cope with this weakness, Wills and Tatar look at many messages for each sub-run and look for a large difference between sub-runs. From the two sub-runs, they construct two distributions over ads. If the distributions are very different, then they suggest that the differences between sensitive information caused differences in the ads shown. They assume that the large number of messages finds patterns beneath the noise of confounding factors, such as ad churn. Formalizing this approach would involve statistics beyond the scope of this paper.

A second approach that Wills and Tatar take to reduce the effects of ad churn is to construct their distributions over ads based not upon the entire content of the ads but rather only upon the topics of the ads (e.g., dogs, cars, golf). In essence, they replaced the set of ads L with a quotient space $L' = L/\equiv$ over ads where $\equiv$ is the "topic equivalence relation" over ads. Using L' reduces the number of positive results since a difference must be large enough to indicate a switch in topics before resulting in different distributions. Thus, small changes from interfering confounding inputs in C will not affect their results. Furthermore, interference from H to L' implies interference from H to L, maintaining soundness.

Completeness. For completeness, we must assume that the analyst selects $\vec{\imath}_1^a$ and $\vec{\imath}_2^a$ that will trigger q's interference if q has interference. These inputs must fully exercise the temptations q might have to use input from H to select outputs in L regardless of the confounding inputs it receives.

Definition 4 (Jointly Exercise)**.** *For an analyst a:A, system q:Q, environment $\vec{e}$:$\vec{E}$, and sets H and L, we say that $\vec{\imath}_1^a$ and $\vec{\imath}_2^a$ being used by A as a differing sting* jointly exercise q for H to L *iff q having interference from H to L implies that $\lfloor Q(\vec{\imath}_1) \downarrow L \rfloor \neq \lfloor Q(\vec{\imath}_2) \downarrow L \rfloor$ where $\vec{m} = [a{:}A || q{:}Q || \vec{e}{:}\vec{E}]$, $\vec{\imath}_1 = \mathcal{U}_1(\vec{m}, \vec{\imath}_1^a)$, and $\vec{\imath}_2 = \mathcal{U}_2(\vec{m}, \vec{\imath}_1^a)$.*

Furthermore, we must assume that if interference occurs, then a can observe the occurrence. We achieve this by assuming that $L \subseteq M_a$, which implies the soundness assumption that $L \cap I_q \subseteq O_a$.

Theorem 2 (Qualified Completeness)**.** *For all analysts a:A, systems q:Q, environments $\vec{e}$:$\vec{E}$, and subsets L and H of M such that A is a differencing sting for H to L using $\vec{\imath}^a = \vec{\imath}_1^a \cdot \vec{\imath}_2^a$ and s to split, if the system q has independent L sub-runs for a providing $\vec{\imath}^a$, s_1 correctly L splits, $\vec{\imath}_1^a$ and $\vec{\imath}_2^a$ jointly exercise q for H to L, and $L \subseteq M_a$, then q having interference from H to L implies a positive result from a.*

Returning to Wills and Tatar's use of the quotient ad space L', a negative result for interference from H to L' does not imply noninterference from H to L since Google could be using the sensitive information in H to select among ads with the same topic. Thus, using L' improves soundness at the expense of completeness.

5.2 Baseline Sting

Guha et al. introduce the baseline sting [2], which is an extended form of differencing that uses additional interactions to convert an assumption needed by differencing into a weaker assumption. The soundness proof for the differencing sting required the analyst to assume independent L sub-runs. Intuitively, the baseline sting starts by testing this assumption and if it appears to hold, runs the differencing sting.

Whereas the differencing sting uses two sub-runs, the baseline sting uses three with inputs $\vec{\imath}^a = \vec{\imath}^a_0 \cdot \vec{\imath}^a_1 \cdot \vec{\imath}^a_2$ to produce the messages $\vec{m}^a = \vec{m}^a_0 \cdot \vec{m}^a_1 \cdot \vec{m}^a_2$. The analyst uses inputs such that $\vec{\imath}^a_0 = \vec{\imath}^a_1$ and $\lfloor \vec{\imath}^a_1 \downarrow L \rfloor = \lfloor \vec{\imath}^a_2 \downarrow L \rfloor$. The first two input sequences $\vec{\imath}^a_0$ and $\vec{\imath}^a_1$ hold a's inputs to q constant to find the baseline amount by which the low-level outputs ($\lfloor \vec{m}^a_0 \downarrow L \rfloor$ and $\lfloor \vec{m}^a_1 \downarrow L \rfloor$ in this case) can change from q's internal state and from inputs to q from other systems. This baseline measures the amount of q's ad churn.

If $\lfloor \vec{m}^a_0 \downarrow L \rfloor$ and $\lfloor \vec{m}^a_1 \downarrow L \rfloor$ differ, then the baseline invalidates either the independent sub-run or noninterference assumption of Theorem 1. If not, the analyst concludes that the assumptions hold and continues with a differencing sting by comparing $\lfloor \vec{m}^a_1 \downarrow L \rfloor$ to $\lfloor \vec{m}^a_2 \downarrow L \rfloor$.

Soundness. The soundness of the differencing sting depends upon the confounding inputs not affecting the low-level outputs to a either by violating the independent sub-run assumption or the noninterference of C to L. The soundness of the baseline sting depends upon any such noise in the system showing up in the baseline comparison of $\lfloor \vec{m}^a_0 \downarrow L \rfloor$ and $\lfloor \vec{m}^a_1 \downarrow L \rfloor$ and causing the sting to abort without a positive result. The sting will be unsound if such noise only starts with the third input sequence $\vec{\imath}^a_2$. Thus, while the differencing sting assumed no such noise, the baseline sting assumes such noise is either always or never present.

For Guha et al.'s TWT study, requiring that baseline shows zero differences (i.e., that $\lfloor \vec{m}^a_0 \downarrow L \rfloor = \lfloor \vec{m}^a_1 \downarrow L \rfloor$) before proceeding to the second step of the sting will result in the second step rarely being reached since Google has significant ad churn [2]. Thus, they employed a relaxed form of the sting that uses long sub-runs in a manner similar to how Wills and Tatar used differencing in practice. Rather than aborting if any difference is detected in the baseline, they use a metric to calculate a baseline amount of difference from ad churn. The second step compares $\vec{\imath}^a_1$ to $\vec{\imath}^a_2$ to determine the amount of change on top of the baseline. If the additional degree of difference is large, then they conclude that the differences between the inputs must have caused interference.

The soundness of the relaxed sting depends upon the amount of noise in the system being constant. If the amount of noise goes up over time, then it will find additional noise for a reasons other than differences in sensitive information.

Guha et al. found Google randomly selecting variations of ads, a form of confounding noise. Thus, they also use a quotient ad space L'. However, theirs differs from Wills and Tatar by focusing the URL displayed by the ad to the user rather than the ad's topic.

While interference from H to L' does imply interference from H to L, less clear are the effects of Guha et al. using the quotient space L' for the baseline sting. In particular, using L' can reduce both the degree of difference computed as a baseline from comparing $\vec{m}^a_0$ to $\vec{m}^a_1$ and degree of

difference from comparing $\vec{m}_1^a$ to $\vec{m}_2^a$. Since the analyst will compare these degrees of difference to one another, the effects can cancel one another out. Thus, it appears that using L' in place of L does not offer the increase in soundness for the baseline sting that it did for the differencing sting.

Completeness. Similar to the differencing sting, if the sting produces a negative result, the assumption that $\vec{\imath}_1^u$ and $\vec{\imath}_2^u$ jointly exercise the system in question q (Def. 4) leads to the conclusion that q has noninterference. However, for completeness, we must also have that the sting does not abort and no noise causes $\lfloor \vec{m}_0^a \downarrow L \rfloor$ and $\lfloor \vec{m}_1^a \downarrow L \rfloor$ to differ.

While Guha et al.'s use of the baseline sting corrects for ad churn to reduce false positives, it has implicit assumptions that could lead to incompleteness. In particular, they state:

> If the ads are identical, we can trivially conclude the ad network doesn't use that user characteristic for targeting

This claim requires an assumption that they fully exercised the functionality of Google. While they took measures to exercise such functionality, we feel that further study is needed on the difficulty of fully exercising services like Google before claims of completeness will be demonstrated.

As with differencing, using a quotient space L' negatively impacts completeness. Indeed, the text of the ads accompanying the displayed URLs can be significantly different. For example, in their paper, they present two ads treated as equivalent in their study. However, only one of them may have targeted the weight conscious ("[...] Fit Every Figure [...]") [2, page 2]. Furthermore, Wills and Tatar found (after Guha et al.'s study) that ads for a single dating site would vary in the images and text surrounding a constant displayed URL to target by sexual orientation [47, Fig. 11], suggesting that using only the displayed URL could miss interference. On the other hand, Balebako et al. analyzed ads using only the displayed URL and using all of the ad's text and found similar results in both cases [4].

We consider further study of how ignoring ad variations affects soundness and completeness to be an important direction for future work. Furthermore, we suggest that analysts consider running two analyses: one with soundness as the goal and one with completeness as the goal, which can often be done on the same collection of data.

Iteration. In practice, the baseline analysis is typically used iteratively. In this case, many inputs $\vec{\imath}^u = \vec{\imath}_1^u \cdot \vec{\imath}_2^u \cdot \vec{\imath}_3^u$ are used by the analyst. Rather than aborting if any difference is detected in the baseline, the iterative version uses a metric to calculate a baseline amount of difference from the noise [2]. The differing step compares $\vec{\imath}_2^u$s to $\vec{\imath}_3^u$s to determine the amount of change on top of the baseline. If the additional degree of difference is large, then the analyst concludes that the differences between the inputs must have caused interference.

The soundness of the iterative analysis depends upon the amount of noise in the system being constant. If the amount of noise can go up from the baseline comparisons to differencing comparisons, the analysis might find additional noise for a reasons other than differences in sensitive information.

5.3 Broadcast Nonce Sting

Recall that the copyright trap involved a publisher placing unusual data into a larger document of common information to detect when a rival publisher copies the document instead of collecting

the information independently. We call the generalization of the copyright trap from the piracy detection application to investigations in general the *broadcast nonce sting* since the unusual data is used as nonce by the analysis and nonce is shared with many systems.

This sting has also been used in TWT. During their study, Wills and Tatar observe Google serving the ad "LGBT for Obama" on `thefreedictionary.com`, a site that is not about LGBT (lesbian, gay, bisexual, or transgendered) issues [3]. While they do not conclude that Google necessarily selects ads based upon a sensitive interest in LGBT issues, they note this behavior as suspicious. We formalize their suspicion with the broadcast nonce sting where, by virtue of being rare, LGBT acts likes a nonce.

An analyst running a broadcast nonce sting embeds a nonce into sensitive information and provides it as output. In Wills and Tatar's study, they visit LGBT-related first-party websites to embed the nonce LGBT into information recording sexual interests. If the analyst observes the nonce in the output of the system in question, the analyst concludes that the system used the sensitive information. In particular, Wills and Tatar examine Google ads hosted on non-LGBT-related sites. They note as suspicious LGBT ads on non-LGBT-related sites since Google possibly selected them based upon the value of information on their sexual interests. That is, LGBT serves to connect Google's selection of a low-level ad to sensitive information provided by browsing LGBT-related sites that are otherwise unrelated to the ad.

To formally define this sting, we use $m \ni n$ to denote that the message m contains the value n, and M_n to denote the set of messages containing n. Intuitively, the analyst uses a value n as a *nonce*. Ideally, no other system can generate n without first receiving n as input as formalized by the next definition. In practice, analysts use unusual values, such as fake addresses, as imperfect nonces.

Definition 5 (Perfect Nonce)**.** *Let $\vec{m}[x]$ be the xth message in $\vec{m}$. For a system a, we call n a* perfect a-nonce *iff for all processes A, all environments $\vec{e}{:}\vec{E}$, all message sequences $\vec{m}$ such that $\vec{m} = [a{:}A||\vec{e}{:}\vec{E}]$, all $j \geq 0$, $\vec{m}[j] \ni n$ implies either (1) $\vec{m}[j]$ is a message from a or (2) that there exists k such that $0 \leq k < j$, $\vec{m}[k] \ni n$ and $\vec{m}[k]$ is a message to the sender of $\vec{m}[j]$.*

For an analyst a to run a broadcast nonce sting, a produces a sequence of outputs at least one of which contains a nonce n. After providing these outputs, a then examines the messages in $O_q \cap I_a$, the outputs of q to a. If any of these messages are also in $L \cap M_n$, then a returns a positive result.

Soundness. The soundness of the broadcast nonce sting depends upon the nonce n being a perfect a-nonce. Furthermore, the broadcast nonce sting is only sound for interference from H to L when H includes every nonce-carrying input to q. For example, Wills and Tatar's observation does not suggest interference if they consider only a subset of the LGBT-related websites they visited to be sensitive since they cannot determine which provided the information to Google. (We consider this problem in Section 5.4.)

Theorem 3 (Qualified Soundness)**.** *For all analysts $a{:}A$, systems $q{:}Q$, environments $\vec{e}{:}\vec{E}$, and sets L and H such that A is a broadcast nonce sting using n on q, if n is a perfect a-nonce and $H \supseteq M_n \cap I_q$, then a returning a positive result implies that q has interference from H to L.*

While reasonable, the assumptions yielding soundness do not strictly hold in practice. The requirement that n is a nonce is unsatisfiable since the system q could generate n by chance. However, the nonce assumption is useful in practice since it can hold with very high certainty if

a sufficiently complex n is selected. Wills and Tatar's use of LGBT as a nonce is reasonable but imperfect. Since only 3.4% of U.S. adults self-identify as LGBT [48], Google selecting LGBT ads without using some information is unlikely since Google will probably aim for the much larger non-LGBT demographic by default.

However, assuming that, without tracking, Google would present ads in proportion to the target population size, we would expect that 3.4% of ads that target a sexual orientation would be LGBT targeting ads. Thus, if the LGBT related ad was only one of a large number of ads targeting sexual orientation, then a conclusion of interference could be a false positive. Analysts can avoid false positives by only producing a positive result upon see many occurrences of the nonce. We recommend that future studies employing imperfect nonces examine whether the system produces the nonce more than would be expected by a system not using the sensitive information.

In practice, some classes of sensitive information, such as gender, has too little entropy to have nonces embedded in them. The differencing and baseline stings are more appropriate for such classes of sensitive information.

Completeness. For completeness, we need that if q has interference, then the sting returns a positive result. Since a positive result is only possible upon seeing a nonce, every sensitive input must contain one, that is, $H \subseteq M_n$. Furthermore, the system q must produce the supplied nonce as output to the analyst provided the input $\vec{\imath}$. This requirement implies that every message in L must be carrying a nonce and be observable to the analyst: $L \subseteq M_n \cap I_a$. Lastly, systems might only leak information under certain conditions. Thus, we also need that the inputs provided by the analyst exercises q in a manner similar to Definition 4:

Definition 6 (Solely Exercises)**.** *For an analyst $a{:}A$, system $q{:}Q$, environment $\vec{e}{:}\vec{E}$, and sets H and L, we say that $\vec{\imath}^{\,u}$ in $(O_a \cap I_q)^*$* solely exercises *q iff interference implies $\lfloor Q(\vec{\imath}){\downarrow}L \rfloor$ contains the nonce n where $\vec{m} = [a{:}A||q{:}Q||\vec{e}{:}\vec{E}]$ and $\vec{\imath} = \mathcal{U}_1(\vec{m}, \vec{\imath}^{\,u})$.*

Theorem 4 (Qualified Completeness)**.** *For all analysts $a{:}A$, systems $q{:}Q$, environments $\vec{e}{:}\vec{E}$, and sets L and H such that A is a broadcast nonce sting using n on q, if $H \subseteq M_n$, $L \subseteq M_n \cap I_a$, and $\vec{\imath}^{\,u}$ solely exercises q, then q having interference from H to L implies a positive result from a.*

Iteration. The analyst can replace the strong assumptions needed for completeness with weaker ones by running multiple analyses and seeing whether any of their results are positive. In particular, the analyst uses a list of nonce analyses and runs one after the next. This reduces the assumption to a weaker one where one of the tests will result in the system providing a nonce as an output.

Furthermore, iteration can reduce the assumptions needed for soundness. In particular, the nonce assumption is strong in that it requires that systems other than a can never produce the nonce independent of receiving it as input. In some cases, the analyst's choice of sensitive inputs to the system might be too limited to supply the system with an appropriate nonce. For example, the analyst might want to test whether a system uses a zip code for producing some output. Since the number of possible zip codes is large but not huge, they can only weakly approximate nonces. However, the analyst can run the nonce analysis multiple times with varying approximate nonces. If many test results are positive, the analyst can conclude with high probability that the system has interference.

5.4 Bilateral Nonce Sting

For TWT, we have been mainly concerned with how the third-party web tracker uses information it collects. However, the analyst might be interested in which first-party websites pass information to the tracker. In some cases, this is easy to determine from network traffic [1]. However, more sophisticated means are needed when the first-party and the tracker communicate using channels unobservable to the analyst, such as when Google pools information from its various first-party web services to create a central profile of a user. Indeed, Wills and Tatar attempted to determine which Google services affect the ads shown to them [3].

In this case, the broadcast nonce analysis will not work since the analyst cannot observe the low-level messages of interest from a first-party website to Google. To handle this case, we need to extend the broadcast nonce sting to use a different nonce for each possible source of Google. We call this extension the *bilateral nonce sting*.

An analyst running a bilateral nonce sting supplies a system in question (e.g., a first-party website) with sensitive information carrying a nonce that the analyst provides to no other system. If the analyst observes the nonce in the output of a third party, the analyst produces a positive result suggesting that the system in question provided the nonce to another system. That is, the sting detects an output of the system to a third party without directly observing it.

Soundness. The soundness of the bilateral nonce sting depends upon the same assumptions as that of the broadcast nonce sting plus the additional assumption that each possible source in question receives a different nonce. This additional assumption allows us to prove that if a only shares the nonce with one possible source and receives the nonce from some third party, then that possible source must have shared the nonce with some (possibly different) third party.

To go from the nonce assumption to a soundness result, we use two lemmas about how nonces can flow in a network. The first shows that if a only shares the nonce with q and receives the nonce from some third party, then q must have further shared the nonce. It allows a to soundly detect a nonce-carrying output from q to a third party without directly observing the output. Since the analyst cannot determine anything else about the output, we require that every such output be considered low-level. That is, we require that $L \supseteq M_n \cap O_q - I_a$ where I_a is exempted since a is not a third party.

The second lemma shows that if a shares a perfect a-nonce only with q, then any inputs to q that contain the nonce will be from a until q further shares the nonce. It is needed to show that the detected output depends upon a high-level input to q. Furthermore, it allows us to relax the requirement in Theorem 3 on H from including every input to q containing a nonce to just including every input to q *from* a containing a nonce.

Theorem 5 (Qualified Soundness)**.** *For all analysts a:A, systems q:Q, environments $\vec{e}$:$\vec{E}$, and sets L and H such that A is a bilateral nonce sting using n on q, if n is a perfect a-nonce, $H \supseteq M_n \cap O_a \cap I_q$, and $L \supseteq M_n \cap O_q - I_a$, then a returning a positive result implies that q has interference from H to L.*

As with the broadcast nonce analysis, finding more nonces in the outputs of a system increases the probability that system copied a nonce containing message even if the nonces are not perfect.

Completeness. For completeness, we must have the inputs exercising the system q and that $H \subseteq M_n$ as in the broadcast nonce sting. We continue to need every message in L to contain a

nonce, but no longer need them to be outputs to a. However, completeness now requires them to to trigger a series of messages that will result in the analyst seeing some message containing the nonce from a system other than q.

As with the broadcast nonce analysis, finding more nonces in the outputs of a system increases the probability that system copied a nonce containing message even if the nonces are not perfect.

5.5 Hybrid Stings

In one part of their study, instead of examining ads, Wills and Tatar examine the outputs of Google's Ad Preference Manager (APM), which lists the interests that Google has inferred about a user. For this study, they implicitly used a differencing sting when they display different sets of interests and then compared the outputs in the APM. Since Google only tracks a limited number of popular interests, interests make for poor nonces. However, due to the large amount of noise in the system, the analysts cannot be sure that differences in the APM are from differences in their behavior. Thus, they only look for differences corresponding to the presence or absence of the interests they displayed during the study. That is, they check whether the output of Google contains values based upon inputs they provide to first-party websites (broadcast nonces) under some profiles but not others (differencing). This *hybrid sting* both accounts for the noisy environment and poor quality of the nonces to improve soundness.

Indeed, in many cases, the analyst's choice of sensitive messages might be too limited embed a strong nonce. In such cases, the analyst can perform multiple sub-runs each conducting a nonce sting using a different weak nonce and examine whether a weak nonce appears more often in the stings using it than the ones not using it.

However, the possibility of hybrid stings does not absolve the analyst for collecting enough information to conduct the analyses. For example, another test of Wills and Tatar involved using LinkedIn and Pandora profiles with the location set to New York City. The authors wanted to determine whether Google used the profile locations for selecting advertisements. However, despite seeing numerous ads for NYC, the authors do not conclude that Google uses the profile location since (1) NYC "is a popular location in general" and (2) they did "not have a baseline for comparison" [47, page 9].

Our framework explains these concerns as threats to soundness. Their first concern is an argument against using NYC as a nonce. Their second concern explains why they could not perform a differencing or baseline sting. Our systematization makes clear what investigations are possible and how to conduct and collect data for each. In particular, it suggests selecting a city less popular than NYC for a nonce and/or collecting additional data without the location of NYC for a differencing, baseline, or hybrid sting.

Another sort of hybrid sting appears in some forms of traitor tracing. For example, in some cases, the analyst must investigate a *pirate decoder* holding illicitly shared cryptographic keys. The analyst uses differencing on the decoder to determine the keys it holds and then reasoning as in nonce analysis, determines which key holders provided keys to the pirate [46].

6 Conclusion and Future Directions

We have identified a range of problems that can be approached systematically as information flow investigations. This work provides a fresh perspective on these problems and on IFA, which has long

been dominated by white box program analysis. We have taken specific stings out of the narrow context for which they were designed and placed them into a general framework. Our framework has allowed us to find the limitations and abilities of investigation in general and of specific stings individually.

In particular, we have examined the emerging area of TWT and formalized studies in the area as stings in our framework. The value of this exercise is two fold. First, by placing these empirical studies into our formal framework of investigations, we can closely study their reasoning. In particular, we discuss whether the implicit assumptions made by these works are reasonable and whether they are using their analyses as effectively as possible.

Second, we test the applicability of noninterference to real problems outside of its comfort zone of program analysis. In particular, we explore applying investigations to real studies. Since our formalism is an abstraction of the actual problems facing researchers and does not, for example, include the statistical inferences that drive empirical science, our examination does not capture every aspect of these studies. Nevertheless, it does elucidate aspects of these studies, such as their soundness and completeness.

This process has lead concrete suggestions and areas of further study:

- *For imperfect nonces, compare the number observed when supplying it to the system to the number expected when not supplying it.* For example, Wills and Tatar's use of LGBT related messages as a nonce is reasonable, but it is not strictly a nonce (Section 5.3). Thus, it would helpful to determine whether such messages appear more often than expected.

- *Let the requirements of a sting guide data collection.* For example, Wills and Tatar's intuitive analysis using New York City did not collect enough for a differencing sting (Section 5.5).

- *Fully exercising systems like Google needs further study before claims of completeness are safe.* Claims such as Guha et al.'s that it is possible to determine that Google does not use information require further study due to the difficulty of fully exercising all of Google's functionality (Section 5.2).

- *Identifying and handling variations in ads from noise requires further research.* Guha et al.'s study raises questions about when two ads should be treated as instances of the same ad (Section 5.2). We consider further study of how ignoring ad variations affects soundness and completeness to be an important direction for future work.

- *Consider running two analyses: one for soundness and one for completeness.* Given the uncertainty surrounding when to consider two ads as the same, we currently must resolve the question by erring on the side of promoting soundness or completeness. Thus, separate analyses may be appropriate for these different goals.

Finally, our work leads to three directions for future work.

Probabilistic Reasoning. A probabilistic framework could systematize statistical inferences from running analyses multiple times or for probabilistic notions of information flow, such as differential privacy [49]. In particular, using statistical properties as a gauge, we could compare the differencing and baseline analyses to experimental designs used in other branches of science (e.g., [50]). Prior work has looked at probabilistic forms of noninterference appropriate for networks of systems

(e.g., [51]). We are interested in how to investigate systems in such settings. For example, we conjuncture that, even without making the nonce assumption, a bilateral nonce analysis can greatly alter the probability that the analyst should assign to the system having interference.

Interrogations and Monitoring. We would like to look at the qualified soundness or completeness of monitoring and interrogation (i.e., non-sting investigations without setups). For example, author de-anonymization (e.g., [52, 53]), detecting cheating (e.g. [54]), and detecting plagiarism of a third-party's work (e.g., [55]) all resemble detecting copyright piracy. However, in these cases, the analyst does not control the sensitive messages (e.g., an anonymous posting) making the setup of embedding nonces impossible. However, in practice, authors are de-anonymized using comparisons. We conjecture that such analyses are sound IFAs under an assumption that no two systems behave too similarly.

Investigations Beyond IFA. Problems outside of IFA are also instances of investigations. For example, Google ran a nonce-like sting to determine whether Bing's search results were mimicking Google's [56]. Thus, rather than tracking information flows, Google's sting involved tracking flows of behavior. In particular, their nonce involved Google returning unusual search results. Google then observed Bing mimicking this behavior after Bing observed users clicking on the unusual results in Internet Explorer.

Another problem is *provenance*, tracking the handling of data [57]. Provenance can be viewed as an extended form of IFA in which the analyst needs to know not just the source of the data, but also the step-by-step flow and handling of the data in a network.

Access-control investigations are also possible. For example, Bowen et al. provide a system of monitored decoy files that attract adversaries into accessing them [58]. Comparing and combining these access-control stings with our flow-based formalism would provide a more comprehensive approach to data governance.

In general, investigations allow the analyst to exercise oversight and detect transgressions by an entity not controlled by the analyst and unwilling 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 while having only limited access to the other. Thus, we envision investigation playing an increasing role in computer security and society in general.

Acknowledgments. We thank Amit Datta, Divya Sharma, and Arunesh Sinha for many helpful comments on this work. We thank the nonce Harry Q. Bovik for inspiration.

References

[1] J. R. Mayer and J. C. Mitchell, "Third-party web tracking: Policy and technology," in *IEEE Symposium on Security and Privacy*, 2012, pp. 413–427.

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

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

[4] 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 Workshop*, 2012.

[5] 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.

[6] N. R. Wagner, "Fingerprinting," in *Proceedings of the 1983 IEEE Symposium on Security and Privacy*, 1983, p. 18.

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

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

[9] Office for Civil Rights, "Summary of the HIPAA privacy rule," OCR Privacy Brief, U.S. Department of Health and Human Services, 2003.

[10] J. A. Goguen and J. Meseguer, "Security policies and security models," in *Proceedings of the IEEE Symposium on Security and Privacy*, 1982, pp. 11–20.

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

[12] D. Milano, "Content control: Digital watermarking and fingerprinting," Rhozet, Harmonic, White Paper. http://www.rhozet.com/whitepapers/Fingerprinting_Watermarking.pdf

[13] N. Patel, "iTunes plus DRM-free, not free of annoying glitches," Engadget webpage, 2007. http://www.engadget.com/2007/05/31/itunes-plus-drm-free-not-free-of-annoying-glitches/

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

[15] RSA Labs, "RSA data loss prevention." 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] M. Arrington, "Orbious will make forwarding confidential documents dangerous," TechCrunch webpage, 2007. http://techcrunch.com/2007/08/05/orbious-will-make-forwarding-confidential-documents-dangerous/

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

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

[20] B. Krishnamurthy, K. Naryshkin, and C. E. Wills, "Privacy leakage vs. protection measures: The growing disconnect," in *Proceedings of the Web 2.0 Security and Privacy Workshop*, 2011, pp. 1–10.

[21] J. T. Haigh and W. D. Young, "Extending the noninterference version of MLS for SAT," *IEEE Trans. Softw. Eng.*, vol. 13, no. 2, pp. 141–150, 1987.

[22] A. W. Roscoe and M. H. Goldsmith, "What is intransitive noninterference?" in *Proceedings of the 12th IEEE Workshop on Computer Security Foundations*, 1999, p. 228.

[23] 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.

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

[25] 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 *Proceedings of the 37th Annual IEEE/ACM International Symposium on Microarchitecture*, 2004, pp. 243–254.

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

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

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

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

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

[31] D. Devriese and F. Piessens, "Noninterference through secure multi-execution," in *Proceedings of the 2010 IEEE Symposium on Security and Privacy*, 2010, pp. 109–124.

[32] R. de Nicola and M. C. B. Hennessy, "Testing equivalences for processes," in *Automata, Languages and Programming*, ser. Lecture Notes in Computer Science, J. Diaz, Ed. Springer Berlin Heidelberg, 1983, vol. 154, pp. 548–560.

[33] ——, "Testing equivalences for processes," *Theoretical Computer Science*, pp. 83–133, 1984.

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

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

[36] M. Y. Vardi, "Automatic verification of probabilistic concurrent finite state programs," in *Proceedings of the 26th Annual Symposium on Foundations of Computer Science*. IEEE Computer Society, 1985, pp. 327–338.

[37] 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.

[38] D. Sutherland, "A model of information," in *Proceedings of the 9th National Computer Security Conference*, 1986.

[39] D. McCullough, "Noninterference and the composability of security properties," in *IEEE Symposium on Security and Privacy*, 1988, pp. 177–186.

[40] J. T. Wittbold and D. M. Johnson, "Information flow in nondeterministic systems," in *Proceedings of the IEEE Symposium on Security and Privacy*, 1990, pp. 144–161.

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

[42] A. Zakinthinos and E. S. Lee, "A general theory of security properties," in *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, 1997, p. 94.

[43] D. Clark and S. Hunt, "Non-interference for deterministic interactive programs," in *Formal Aspects in Security and Trust*, P. Degano, J. Guttman, and F. Martinelli, Eds. Springer-Verlag, 2009, pp. 50–66.

[44] D. M. Volpano, "Safety versus secrecy," in *Proceedings of the 6th International Symposium on Static Analysis*. Springer-Verlag, 1999, pp. 303–311.

[45] G. R. Newman and K. Socia, "Sting operations," U.S. Department of Justice, Response Guides 6, 2007.

[46] D. Boneh and M. Naor, "Traitor tracing with constant size ciphertext," in *Proceedings of the 15th ACM Conference on Computer and Communications Security*, 2008, pp. 501–510.

[47] C. E. Wills and C. Tatar, "Understanding what they do with what they know," Computer Science Department, Worcester Polytechnic Institute, Tech. Rep. WPI-CS-TR-12-03, 2012.

[48] G. J. Gates and F. Newport, "3.5% of U.S. adults identify as LGBT: Inaugural Gallup finding based on more than 120,000 interviews," Gallup, Special Report, 2012.

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

[50] R. A. Bailey, *Design of Comparative Experiments*. Cambridge University Press, 2008.

[51] M. Backes and B. Pfitzmann, "Intransitive non-interference for cryptographic purposes," in *Proceedings of the 2003 IEEE Symposium on Security and Privacy*, 2003, p. 140.

[52] E. Stamatatos, "A survey of modern authorship attribution methods," *J. Am. Soc. Inf. Sci. Technol.*, vol. 60, no. 3, pp. 538–556, 2009.

[53] A. Narayanan, H. Paskov, N. Z. Gong, J. Bethencourt, E. Stefanov, E. C. R. Shin, and D. Song, "On the feasibility of internet-scale author identification," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, 2012, pp. 300–314.

[54] D. J. Palazzo, Y.-J. Lee, R. Warnakulasooriya, and D. E. Pritchard, "Patterns, correlates, and reduction of homework copying," *Phys. Rev. ST Phys. Educ. Res.*, vol. 6, p. 010104, 2010.

[55] H. Maurer, F. Kappe, and B. Zaka, "Plagiarism – a survey," *Journal of Universal Computer Science*, vol. 12, no. 8, p. 1050, 2006.

[56] D. Sullivan, "Bing: Why Googles wrong in its accusations," Search Engine Land blog, 2011. http://searchengineland.com/bing-why-googles-wrong-in-its-accusations-63279

[57] P. Buneman, S. Khanna, and W. C. Tan, "Why and where: A characterization of data provenance," in *Proceedings of the 8th International Conference on Database Theory*. Springer-Verlag, 2001, pp. 316–330.

[58] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, "Baiting inside attackers using decoy documents," in *SecureComm*, 2009, pp. 51–70.

A Proofs for the Differencing Sting

We use lemmas to prove the qualified soundness of the differencing sting. The first shows the correctness of s_1 for splitting implies that s_2 is also correct.

Lemma 1. *For all analysts $a{:}A$, systems $q{:}Q$, environments $\vec{e}{:}\vec{E}$, sets L and H, input sequences $\vec{\imath}^a = \vec{\imath}^a_1{\cdot}\vec{\imath}^a_2$, and splitting functions s, if $\vec{\imath}^a = \lfloor \vec{m}{\downarrow}O_a \cap I_q \rfloor$, the system q has independent L sub-runs for a providing $\vec{\imath}^a$, s_1 correctly and L splits, then $\lfloor s_2(\vec{m}^a){\downarrow}L \rfloor = \lfloor Q(\vec{\imath}_2){\downarrow}L \cap M_a \rfloor$ where $\vec{m} = [a{:}A||q{:}Q||\vec{e}{:}\vec{E}]$, $\vec{m}^a = \lfloor \vec{m}{\downarrow}M_a \rfloor$, and $\vec{\imath}_2 = \mathcal{U}_2(\vec{m}, \vec{\imath}^a_1)$.*

Proof. Assume all of the conditions of the lemma. Let $\vec{m}^{qa}_1 = s_1(\vec{m}^a)$ and $\vec{m}^{qa}_2 = s_2(\vec{m}^a)$. Let $\vec{\imath} = \lfloor \vec{m}{\downarrow}I_q \rfloor$.

Since $\vec{\imath} = \lfloor \vec{m}{\downarrow}I_q \rfloor$ and no message can be an input to more than one system, none of the messages in $\vec{\imath}$ can be inputs to a. Since $\vec{\imath}^a = \lfloor \vec{m}{\downarrow}O_a \cap I_q \rfloor$, $\lfloor \vec{\imath}{\downarrow}M_a \rfloor = \lfloor \lfloor \vec{m}{\downarrow}I_q \rfloor {\downarrow}M_a \rfloor = \vec{\imath}^a_1{\cdot}\vec{\imath}^a_2$.

Let $\vec{\imath}_1$ be $\mathcal{U}_1(\vec{m}, \vec{\imath}^a_1)$. Since $\vec{\imath} = \lfloor \vec{m}{\downarrow}I_q \rfloor$, $\vec{\imath}_1$ is a prefix of $\lfloor \vec{m}{\downarrow}I_q \rfloor$. Let $\vec{\imath}_2$ be $\mathcal{U}_2(\vec{m}, \vec{\imath}^a_1)$. Thus, $\vec{\imath} = \vec{\imath}_1{\cdot}\vec{\imath}_2$.

Since q has independent L sub-runs, we know that $\lfloor Q(\vec{\imath}_1{\cdot}\vec{\imath}_2){\downarrow}L \cap M_a \rfloor = \lfloor Q(\vec{\imath}_1){\downarrow}L \cap M_a \rfloor {\cdot} \lfloor Q(\vec{\imath}_2){\downarrow}L \cap M_a \rfloor$. Since s_1 correctly L splits, $\lfloor Q(\vec{\imath}_1){\downarrow}L \cap M_a \rfloor = \lfloor s_1(\vec{m}^a){\downarrow}L \rfloor = \lfloor \vec{m}^{qa}_1{\downarrow}L \rfloor$. By the definition of s_2,

$\lfloor \vec{m}^a \downarrow M_q \rfloor = m_1^{qa} \cdot m_2^{qa}$. Thus,

$$\begin{aligned}
&\lfloor Q(\vec{\imath}_1)\downarrow L \cap M_a \rfloor \cdot \lfloor Q(\vec{\imath}_2)\downarrow L \cap M_a \rfloor \\
&\qquad = \lfloor Q(\vec{\imath}_1 \cdot \vec{\imath}_2)\downarrow M_a \cap L \rfloor \\
&\qquad = \lfloor Q(\lfloor \vec{m}\downarrow I_q \rfloor)\downarrow M_a \cap L \rfloor \\
&\qquad = \lfloor \vec{m}\downarrow M_a \cap M_q \cap L \rfloor \\
&\qquad = \lfloor \lfloor \lfloor \vec{m}\downarrow M_a \rfloor \downarrow M_q \rfloor \downarrow L \rfloor \\
&\qquad = \lfloor \lfloor \vec{m}^a \downarrow M_q \rfloor \downarrow L \rfloor \\
&\qquad = \lfloor m_1^{qa} \cdot m_2^{qa} \downarrow L \rfloor \\
&\qquad = \lfloor m_1^{qa} \downarrow L \rfloor \cdot \lfloor m_2^{qa} \downarrow L \rfloor \\
&\qquad = \lfloor Q(\vec{\imath}_1)\downarrow L \cap M_a \rfloor \cdot \lfloor m_2^{qa} \downarrow L \rfloor
\end{aligned}$$

Thus, $\lfloor s_2(m^a)\downarrow L \rfloor = \lfloor m_2^{qa} \downarrow L \rfloor = \lfloor Q(\vec{\imath}_2)\downarrow L \cap M_a \rfloor$. □

Lemma 2. *For all analysts a:A, systems q:Q, and environments $\vec{e}$:$\vec{E}$; for all partitions L, C, and H of M; for all $\vec{\imath}_1$ and $\vec{\imath}_2$ in I_q^*: if q has noninterference from C to L and $\lfloor \vec{\imath}_1\downarrow L \rfloor = \lfloor \vec{\imath}_2\downarrow L \rfloor$, then there exists $\vec{\imath}_3$ such that $\lceil \vec{\imath}_3\uparrow H \rceil = \lceil \vec{\imath}_1\uparrow H \rceil$, $\lceil \vec{\imath}_3\uparrow C \rceil = \lceil \vec{\imath}_2\uparrow C \rceil$, and $\lfloor Q(\vec{\imath}_3)\downarrow L \rfloor = \lfloor Q(\vec{\imath}_2)\downarrow L \rfloor$.*

Proof. Let $\vec{\imath}_3 = \mathsf{i3}(\vec{\imath}_1, \vec{\imath}_2)$ where:

$$\begin{aligned}
\mathsf{i3}(\ell\cdot\vec{m}_1, \ell\cdot\vec{m}_2) &= \ell\cdot\mathsf{i3}(\vec{m}_1, \vec{m}_2) && \text{where } \ell \in L && \text{(keep)} \\
\mathsf{i3}(\vec{m}_1, h\cdot\vec{m}_2) &= h\cdot\mathsf{i3}(\vec{m}_1, \vec{m}_2) && \text{where } h \in H && \text{(keep)} \\
\mathsf{i3}(c\cdot\vec{m}_1, \vec{m}_2) &= c\cdot\mathsf{i3}(\vec{m}_1, \vec{m}_2) && \text{where } c \in C && \text{(keep)} \\
\mathsf{i3}(h\cdot\vec{m}_1, \vec{m}_2) &= \mathsf{i3}(\vec{m}_1, \vec{m}_2) && \text{where } h \in H && \text{(drop)} \\
\mathsf{i3}(\vec{m}_1, c\cdot\vec{m}_2) &= \mathsf{i3}(\vec{m}_1, \vec{m}_2) && \text{where } c \in C && \text{(drop)}
\end{aligned}$$

where the first applicable line is used if more than one applies. Since the only case applying to L requires that both input sequences have the same first low-level inputs ℓ, $\mathsf{i3}(\vec{\imath}_1, \vec{\imath}_2)$ completely defined only if $\lfloor \vec{\imath}_1\downarrow L \rfloor = \lfloor \vec{\imath}_2\downarrow L \rfloor$, which is the case in this proof.

By construction, $\mathsf{i3}(\vec{\imath}_1, \vec{\imath}_2)$ keeps every input in L or C from $\vec{\imath}_1$. Thus, $\lfloor \vec{\imath}_3\downarrow L \cup C \rfloor = \lfloor \vec{\imath}_1\downarrow L \cup C \rfloor$, which implies that $\lceil \vec{\imath}_3\uparrow H \rceil = \lceil \vec{\imath}_1\uparrow H \rceil$ since L, C, and H partition M.

Furthermore, $\mathsf{i3}(\vec{\imath}_1, \vec{\imath}_2)$ keeps every input in L or H from $\vec{\imath}_2$. Thus, $\lfloor \vec{\imath}_3\downarrow L \cup H \rfloor = \lfloor \vec{\imath}_2\downarrow L \cup H \rfloor$, which implies that $\lceil \vec{\imath}_3\uparrow C \rceil = \lceil \vec{\imath}_2\uparrow C \rceil$. Since C does not interfere with L, $\lceil \vec{\imath}_3\uparrow C \rceil = \lceil \vec{\imath}_2\uparrow C \rceil$ implies that $\lfloor Q(\vec{\imath}_3)\downarrow L \rfloor = \lfloor Q(\vec{\imath}_2)\downarrow L \rfloor$. □

Proof of Theorem 1 (Qualified Soundness). Assume all of the conditions of the theorem. Let $\vec{m}^a = \lfloor \vec{m}\downarrow M_a \rfloor$, $\vec{m}_1^{qa} = s_1(\vec{m}^a)$, $\vec{m}_2^{qa} = s_2(\vec{m}^a)$, $\vec{\imath} = \lfloor \vec{m}\downarrow I_q \rfloor$, $\vec{\imath}_1 = \mathcal{U}_1(\vec{m}, \vec{\imath}_1^u)$, and $\vec{\imath}_2 = \mathcal{U}_2(\vec{m}, \vec{\imath}_1^u)$.

Since q has independent L sub-runs, $\lfloor Q(\vec{\imath}_1\cdot\vec{\imath}_2)\downarrow L \cap M_a \rfloor = \lfloor Q(\vec{\imath}_1)\downarrow L \cap M_a \rfloor \cdot \lfloor Q(\vec{\imath}_2)\downarrow L \cap M_a \rfloor$. Since s correctly L splits, $\lfloor Q(\vec{\imath}_1)\downarrow L \cap M_a \rfloor = \lfloor \vec{m}_1^{qa}\downarrow L \rfloor$. By Lemma 1, $\lfloor Q(\vec{\imath}_2)\downarrow L \cap M_a \rfloor = \lfloor \vec{m}_2^{qa}\downarrow L \rfloor$.

Since a has a positive result, it must be the case that

$$\lfloor Q(\vec{\imath}_1)\downarrow L \cap M_a \rfloor = \lfloor \vec{m}_1^{qa}\downarrow L \rfloor \neq \lfloor \vec{m}_2^{qa}\downarrow L \rfloor = \lfloor Q(\vec{\imath}_2)\downarrow L \cap M_a \rfloor$$

Thus, since $L \supseteq L \cap M_a$, $\lfloor Q(\vec{\imath}_1)\downarrow L \rfloor \neq \lfloor Q(\vec{\imath}_2)\downarrow L \rfloor$.

Since $\lfloor \vec{\imath}_1^{\,a} \downarrow L \rfloor = \lfloor \vec{\imath}_2^{\,a} \downarrow L \rfloor$ and $L \cap I_q \subseteq O_a$,

$$\lfloor \vec{\imath}_1 \downarrow L \rfloor = \lfloor \lfloor \vec{\imath}_1 \downarrow M_a \rfloor \downarrow L \rfloor = \lfloor \vec{\imath}_1^{\,a} \downarrow L \rfloor = \lfloor \vec{\imath}_2^{\,a} \downarrow L \rfloor = \lfloor \lfloor \vec{\imath}_2 \downarrow M_a \rfloor \downarrow L \rfloor = \lfloor \vec{\imath}_2 \downarrow L \rfloor$$

Since C does not interfere with L and $\lfloor \vec{\imath}_1 \downarrow L \rfloor = \lfloor \vec{\imath}_2 \downarrow L \rfloor$, there exists $\vec{\imath}_3$ such that $\lceil \vec{\imath}_3 \uparrow H \rceil = \lceil \vec{\imath}_1 \uparrow H \rceil$ and $\lfloor Q(\vec{\imath}_3) \downarrow L \rfloor = \lfloor Q(\vec{\imath}_2) \downarrow L \rfloor$ by Lemma 2. Thus, $\lfloor Q(\vec{\imath}_1) \downarrow L \rfloor \neq \lfloor Q(\vec{\imath}_2) \downarrow L \rfloor = \lfloor Q(\vec{\imath}_3) \downarrow L \rfloor$.

Since $\lceil \vec{\imath}_1 \uparrow H \rceil = \lceil \vec{\imath}_3 \uparrow H \rceil$ and $\lfloor Q(\vec{\imath}_1) \downarrow L \rfloor \neq \lfloor Q(\vec{\imath}_3) \downarrow L \rfloor$, $\vec{\imath}_1$ and $\vec{\imath}_3$ show that q has interference from H to L. □

Proof of Theorem 2 (Qualified Completeness). Assume all of the conditions of the theorem. Let $\vec{m} = [a{:}A || q{:}Q || \vec{e}{:}\vec{E}]$. Let $\vec{m}_1^{\,qa} = s_1(\lfloor \vec{m} \downarrow M_a \rfloor)$ and $\vec{m}_2^{\,qa} = s_2(\lfloor \vec{m} \downarrow M_a \rfloor)$. Let $\vec{\imath} = \lfloor \vec{m} \downarrow I_q \rfloor$. Let $\vec{\imath}_1$ be $\mathcal{U}_1(\vec{m}, \vec{\imath}_1^{\,a})$. Let $\vec{\imath}_2$ be $\mathcal{U}_2(\vec{m}, \vec{\imath}_1^{\,a})$.

If q has interference from H to L, then $\lfloor Q(\vec{\imath}_1) \downarrow L \rfloor \neq \lfloor Q(\vec{\imath}_2) \downarrow L \rfloor$ by the assumption that $\vec{\imath}_1^{\,a}$ and $\vec{\imath}_2^{\,a}$ jointly exercise q. Thus, $\lfloor Q(\vec{\imath}_1) \downarrow L \cap M_a \rfloor \neq \lfloor Q(\vec{\imath}_2) \downarrow L \cap M_a \rfloor$ since $L \subseteq M_a$ implies that $L \cap M_a = L$.

Since s correctly L splits and q has independent L sub-runs, $\lfloor Q(\vec{\imath}_1) \downarrow L \cap M_a \rfloor = \lfloor \vec{m}_1^{\,a} \downarrow L \rfloor$. Furthermore, by Lemma 1, $\lfloor Q(\vec{\imath}_2) \downarrow L \cap M_a \rfloor = \lfloor \vec{m}_2^{\,a} \downarrow L \rfloor$. Thus, $\lfloor \vec{m}_1^{\,qa} \downarrow L \cap M_a \rfloor \neq \lfloor \vec{m}_2^{\,qa} \downarrow L \cap M_a \rfloor$. This means that a will return a negative result. □

B Proofs for the Broadcast Nonce Sting

Proof of Theorem 3 (Qualified Soundness). Assume that n is an a-nonce and that a is running such a broadcast nonce sting and that it returns a positive result. If this is the case, then a and q must be in some environment $\vec{e}{:}\vec{E}$. Let $\vec{m} = [a{:}A || q{:}Q || \vec{e}{:}\vec{E}]$.

Since the sting has a positive result, it must be the case that some input to a from q contains n. Thus, there must exist some j that is the first j such that $\vec{m}[j] \in I_a \cap O_q$ and $\vec{m}[j] \ni n$. Since n is an a-nonce and $\vec{m}[j]$ is not from a, it must be the case that there exists k such that $0 \leq k < j$, $\vec{m}[k] \ni n$, and $\vec{m}[k]$ is a message to q.

Let $\vec{\imath}_1 = \lfloor \vec{m} \downarrow I_q \rfloor$. By definition, $\lfloor \vec{m} \downarrow M_q \rfloor = Q(\vec{\imath}_1)$. Since $\vec{\imath}_1$ includes all the inputs to q, the input $\vec{m}[k]$ must be in $\vec{\imath}_1$.

Let $\vec{\imath}_2 = \lfloor \vec{\imath}_1 \downarrow L \rfloor$. Since all the nonce containing inputs of $\vec{\imath}_1$ are in H, $\vec{\imath}_2$ will contain no such inputs. Thus, $Q(\vec{\imath}_2)$ cannot produce an output containing n. However, $Q(\vec{\imath}_1)$ did contain such an output: $\vec{m}[k]$. Thus, $Q(\vec{\imath}_1) \neq Q(\vec{\imath}_2)$. However,

$$\lfloor \vec{\imath}_1 \downarrow L \rfloor = \lfloor \lfloor \vec{\imath}_1 \downarrow L \rfloor \downarrow L \rfloor = \lfloor \vec{\imath}_2 \downarrow L \rfloor$$

This proves that i_1 and i_2 exhibit of the interference of $q{:}Q$ from H to L. □

Proof of Theorem 4 (Qualified Completeness). Assume all of the conditions of the theorem. Since $\vec{\imath}$ solely exercises q, if q has interference than the nonce will appear in $\lfloor Q(\vec{\imath}) \downarrow L \rfloor$. Thus, there exists a message m^n in $\lfloor Q(\vec{\imath}) \downarrow L \rfloor$ that contains the nonce n. Since m^n is in $\lfloor Q(\vec{\imath}) \downarrow L \rfloor$, m^n must be in the sequence $Q(\vec{\imath})$ and, thus, in the set M_q. Furthermore, m^n must be a member of the set L.

Since $L \subseteq M_n \cap I_a$, m^n must be in $M_n \cap I_a$. Since $m^n \in I_a$, m^n cannot be in I_q. Thus, since m^n is in $M_q = I_q \cup O_q$, m^n must be in O_q.

Thus, m^n is in $O_q \cap I_a \cap L \cap M_n$ and the analysis will produce a positive result. □

C Proof of Qualified Soundness for Bilateral Nonce Sting

Lemma 3. *For all $a{:}A$, $q{:}Q$, $\vec{e}{:}\vec{E}$, for all $j \leq |\vec{m}|$, for all s in $\vec{e}$ such that $s \neq a$ and $s \neq q$, and for all r in $\vec{e}$ such that $r \neq q$ if $\lfloor \vec{m}{\downarrow}O_a - I_q \rfloor \not\ni n$, $\vec{m}[j] \in O_s \cap I_r$, and $\vec{m}[j] \ni n$, then there must exist $k < j$ and ℓ in $\vec{e}$ such that $\ell \neq a$, $\ell \neq q$, $\vec{m}[k] \in O_q \cap I_\ell$, and $\vec{m}[k] \ni n$ where $\vec{m} = [a{:}A||q{:}Q||\vec{e}{:}\vec{E}]$.*

Proof. Proof by induction on j.

Base Case: $j = 0$. Since n is an a-nonce, $\vec{m}[0] \ni n$, and $\vec{m}[0] \in O_s$, and $s \neq a$, it must be the case there exists k' such that $0 \leq k' < 0$ such that various conditions hold on k'. However, no such k' can exist since $0 \leq k' < 0$ cannot be satisfied. Thus, we have a contradiction and this case trivially holds.

Inductive Case: Since n is an a-nonce, $\vec{m}[j] \ni n$, and $\vec{m}[j] \in O_s$, and $s \neq a$, case (2) of Definition 5 must hold: there exists j' such that $0 \leq j' < j$, $\vec{m}[j'] \ni n$, and $\vec{m}[j']$ is a message to the sender of $\vec{m}[j]$. Let s' be the sender of $\vec{m}[j']$ to s. That is, $\vec{m}[j'] \in O_{s'} \cap I_s$. We consider three cases:

- $s' = a$. Since $\lfloor \vec{m}{\downarrow}O_a - I_q \rfloor \not\ni n$ and $\vec{m}[j']$ is sent to $s \neq q$, we have a contradiction. Thus, the result trivially holds.
- $s' = q$. If we let $k = j'$ and $\ell = s$, then $\vec{m}[k] \in O_q \cap I_\ell$ since $s' = q$. Furthermore, $\vec{m}[k] \ni n$. Thus, we have shown the needed result.
- $s' \neq q$ and $s' \neq a$. In this case, since $j' < j$, $s \neq q$, $\vec{m}[j'] \in O_{s'} \cap I_s$, and $\vec{m}[j'] \ni n$, we can use the inductive hypothesis with j' for j, s' for s, and s for r. This application of the inductive hypothesis implies that there must exist $k' < j'$ and ℓ' in $\vec{e}$ such that $\ell' \neq a$, $\ell' \neq q$, $\vec{m}[k'] \in O_q \cap I_{\ell'}$, and $\vec{m}[k'] \ni n$.

  If we let k be k' and ℓ be ℓ', then there exists $k = k' < j' < j$ and ℓ in $\vec{e}$ such that $\ell \neq a$, $\ell \neq q$, $\vec{m}[k] \in O_q \cap I_\ell$, and $\vec{m}[k] \ni n$ as needed.

□

Lemma 4. *For all $a{:}A$, $q{:}Q$, $\vec{e}{:}\vec{E}$, if n is a a-nonce; $\lfloor \vec{m}{\downarrow}O_a - I_q \rfloor \not\ni n$; and k is the least k such that there exists ℓ in $\vec{e}$ such that $\ell \neq a$, $\ell \neq q$, $\vec{m}[k] \in O_q \cap I_\ell$, and $\vec{m}[k] \ni n$; then for all $k' < k$, $\vec{m}[k'] \ni n$ implies either $\vec{m}[k'] \in O_q \cap I_a$ or $\vec{m}[k'] \in O_a \cap I_q$ where $\vec{m} = [a{:}A||q{:}Q||\vec{e}{:}\vec{E}]$.*

Proof. Proof by induction over k'.

Base Case: $k' = 0$. Assume $\vec{m}[0] \ni n$. Then since n is an a-nonce either case (1) or case (2) must hold of Definition 5. However, case (2) cannot hold since no messages can come before $\vec{m}[0]$. Thus, case (1) must hold and $\vec{m}[0] \in O_a$. Since a only sends n to q (and self-messages are disallowed), $\vec{m}[0]$ must also be in I_q.

Inductive Case: Assume $\vec{m}[k'] \ni n$. Let the receiver of $\vec{m}[k']$ be r and the sender s. That is, $\vec{m}[k'] \in O_s \cap I_r$. Since n is an a-nonce, then either case (1) or case (2) of Definition 5 holds. We consider each case:

- In case (1), $\vec{m}[k'] \in O_a$. Since $\vec{m}[k']$ is in O_a and O_s, $a = s$. Since a only sends the nonce to q, $\vec{m}[k']$ is in $O_a \cap I_q$ as needed.

- In case (2), there exists $k'' < k'$ such that $\vec{m}[k''] \ni n$ and $\vec{m}[k''] \in O_t \cap I_s$ where the sender of $\vec{m}[k'']$ is t such that $t \neq s$. We may further assume that $s \neq a$, since case (1) handles that possibility.

  Since $k'' < k'$ and $\vec{m}[k''] \ni n$, either $\vec{m}[k''] \in O_q \cap I_a$ or $\vec{m}[k''] \in O_a \cap I_q$ by the inductive hypothesis. Since $s \neq a$, it must be the case that $\vec{m}[k''] \in O_a \cap I_q$. That is, $s = q$ and $t = a$. Thus, $\vec{m}[k'] \in O_q$.

  Since $k' < k$ and k is the least k such that there exists ℓ in $\vec{e}$ such that $\ell \neq a$, $\ell \neq q$, $\vec{m}[k] \in O_q \cap I_\ell$, and $\vec{m}[k] \ni n$, it must be the case that $r = a$ or else r would imply that k' would be an even smaller such k with r playing the role of ℓ. Thus, $\vec{m}[k'] \in O_q \cap I_a$ as needed.

□

Proof of Theorem 5 (Qualified Soundness). Assume that n is an a-nonce and that a is running such a bilateral nonce sting and that it returns a positive result. If this is the case, then a and q must be in some environment $\vec{e}{:}\vec{E}$. Let $\vec{m} = [a{:}A||q{:}Q||\vec{e}{:}\vec{E}]$.

Since the sting has a positive result, it must be the case that some input to a from a system other than q contains n. Thus, there must exist some j that is the least j such that $\vec{m}[j] \in I_a - O_q$ and $\vec{m}[j] \ni n$. Let $s \neq q$ be the sender of $\vec{m}[j]$.

We may use Lemma 3 with $r = a$ to find that there must exist $k < j$ and ℓ in $\vec{e}$ such that $\ell \neq a$, $\ell \neq q$, $\vec{m}[k] \in O_q \cap I_\ell$, and $\vec{m}[k] \ni n$. If more than one such k and ℓ exists, we take k and ℓ to be those with the lowest value of k.

For this k, Lemma 4 applies: for all $k' < k$, if $\vec{m}[k'] \ni n$, then either $\vec{m}[k'] \in O_q \cap I_a$ or $\vec{m}[k'] \in O_a \cap I_q$.

Let $\vec{\imath}_1 = \lfloor \vec{m}{\downarrow}I_q \rfloor$. By definition, $\lfloor \vec{m}{\downarrow}M_q \rfloor = Q(\vec{\imath}_1)$. Since n is an a-nonce and q produces an output $\vec{m}[k] \ni n$, it must be the case that q receives n as part of an input i^n before $\vec{m}[k]$. Since $\vec{\imath}_1$ includes all the inputs to q, the input i^n must be in $\vec{\imath}_1$. Furthermore, as shown with Lemma 4, this input i^n and all others containing the nonce and coming before $\vec{m}[k]$ must be from a. Thus, they are all in $O_a \cap I_q \cap M_n$ and therefore in H.

Let $\vec{\imath}_2 = \lfloor \vec{\imath}_1{\downarrow}L \rfloor$. Since all the nonce containing inputs of $\vec{\imath}_1$ coming before $\vec{m}[k]$ are in H, $\vec{\imath}_2$ will contain no such inputs. Thus, $Q(\vec{\imath}_2)$ cannot produce an output containing n before receiving one from some system other than a. However, $Q(\vec{\imath}_1)$ did contain such an output: $\vec{m}[k]$. Thus, $Q(\vec{\imath}_1) \neq Q(\vec{\imath}_2)$. However,

$$\lfloor \vec{\imath}_1{\downarrow}L \rfloor = \lfloor \lfloor \vec{\imath}_1{\downarrow}L \rfloor {\downarrow}L \rfloor = \lfloor \vec{\imath}_2{\downarrow}L \rfloor$$

This proves that i_1 and i_2 exhibit of the interference of $q{:}Q$ from H to L. □