NPS-CS-09-007

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

IS CROSS-DOMAIN FAULT LOCALIZATION FEASIBLE?

by

William D. Fischer Geoffrey G. Xie Joel D. Young

February 2009

Approved for public release; distribution is unlimited
This report was prepared for and funded by the National Science Foundation, Arlington, VA

THIS PAGE INTENTIONALLY LEFT BLANK

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

Daniel T. Oliver
President

Leonard A. Ferrari
Executive Vice President and
Provost

This report was prepared for the Naval Postgraduate School and funded by the National Science Foundation.

This report was prepared by:

William D. Fischer
LTC

Geoffrey G. Xie
Professor

Joel D. Young
LTC, Assistant Professor

Reviewed by:

Peter J. Denning
Chair, Department of Computer Science

Released by:

Karl A. van Bibber
Vice President and Dean of Research

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE *Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** February 2009	**3. REPORT TYPE AND DATES COVERED** Technical Report

4. TITLE AND SUBTITLE: Title (Mix case letters)
Is Cross-Domain Fault Localization Feasible?

5. FUNDING NUMBERS
CNS-0520210
CNS-0721574

6. AUTHOR(S) William D. Fischer, Geoffrey G. Xie, Joel D. Young

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER NPS-CS-09-007

9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Science Foundation, 4201 Wilson Blvd, Arlington, VA 22230

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES The views expressed in this report are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT (maximum 200 words)

Troubleshooting network faults is challenging. It is even more difficult to diagnose cross-domain issues without complete knowledge of observations and topology from neighboring network domains. For both competitive and security reasons, domain managers hesitate to share observations even when doing so may significantly ease fault localization. In this paper, we present the first comprehensive evaluation of the feasibility of cross-domain fault localization. Leveraging a recent graph-digest based formulation of the problem, we have developed a set of practical metrics and performed extensive experiments to evaluate two key questions: Does cross-domain fault localization offer the kinds of benefits warranting further research? Can it provide deployable and acceptable privacy protection with manageable complexity? We evaluate both provider-customer and peering relationship scenarios between small, medium, and large domains. Our results show that cross-domain fault localization is effective using the graph-digest approach and that sensitive network connectivity properties can be concealed from other domains.

14. SUBJECT TERMS
Network, Networking, Fault Localization, Cross-Domain, Cross Domain, Bayesian, Artificial Intelligence

15. NUMBER OF PAGES 54

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

When observations arising from a network fault propagate across domain boundaries, the fault is described as *cross-domain*. Troubleshooting faults is a challenging task—it is even more difficult when trying to troubleshoot cross-domain issues without knowledge of fault observations and network structure from neighboring network domains. Acquiring knowledge of the needed observations and network topology is complicated by the fact that it is risky, for both competitive and security reasons, for domain managers to share this information even when the sharing might ease fault localization. With business processes migrating to web-services, implemented in the "cloud" and built on protocols such as SOAP (Simple Object Access Protocol), the likelihood of network faults impacting multiple domains approaches unity. We see a dramatic need for methods enabling cross-domain fault localization efficiently while minimizing the need to share sensitive proprietary information.

Consider the simple failure scenario depicted in Fig. 1. A work group in Domain 1 must access data from a server in Domain 3 requiring connectivity through Domain 2. Unfortunately, one of the routers in Domain 2 is misconfigured. Other groups and services can reach the server in Domain 3, but users in Domain 1's work group can't. Furthermore, no equipment failures along the path from the work group (Domain 1) to the server (Domain 3) trigger alarms. This is difficult to troubleshoot without cross-domain collaboration, often resulting in "finger pointing."

Three approaches are possible for diagnosing cross-domain problems. The first of these, the status quo, is *isolated inference*. In this approach, each domain tries to locate the fault without sharing data with other domains. The second approach, which we refer to as *full disclosure*, entails full collaboration and data-sharing between domains. An inference graph using this approach is equivalent to a global graph of all domains involved. While full-disclosure, in general, is unrealistic because of the privacy factor and for scalability reasons, we include it as a baseline model for studying

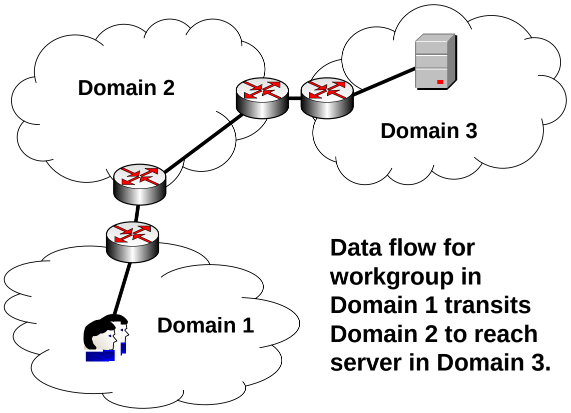
Domain 2
Domain 3
Domain 1
Data flow for workgroup in Domain 1 transits Domain 2 to reach server in Domain 3.

Figure 1. Simple Failure Scenario

inference gains achievable from information sharing. The third approach is *privacy preserving* collaboration. In this approach domains exchange limited information, e.g. summaries of fault observations, to perform inference while protecting sensitive information.

Recently, Fischer *et al* [4] proposed the first concrete solution framework for the *privacy preserving* approach. The framework allows network administrators to collaboratively perform *probabilistic* cross-domain fault localization while preserving privacy by sharing summarized network *causal graphs*. The summaries are called *graph-digests*. Fischer *et al* presented a simple scenario to illustrate that these graph-digests can be carefully crafted to protect sensitive network properties while enhancing the quality of fault localization.

However, Fischer *et al* gave neither a thorough experimental evaluation of their approach nor an evaluation of the trade-offs involved in maintaining accurate cross-domain inference while preserving privacy. In this paper, we provide the first comprehensive evaluation of the feasibility of cross-domain fault localization, focusing on two key questions:

1. *Does cross-domain fault localization offer the kinds of benefits warranting further research?* If the benefits are marginal, there is no reason for operators to risk leaking sensitive information.

2. *Can cross-domain fault localization provide deployable and acceptable privacy protection with manageable complexity?* Cross-domain fault localization should only be deployed if an acceptable level of privacy protection can be provided with manageable complexity. Operators rightly hesitate to adopt a solution before understanding the associated security risks.

In the process of answering these questions, we make the following contributions:

- We have developed a systematic methodology for evaluating the feasibility of a digest-graph based system. The heart of our method is a complete set of metrics for quantifying the performance of a particular digest-graph based design. It is applicable to any network scenario, any cross-domain fault localization algorithm, any alternate graph-digest approach, and any technique developed for eliciting sensitive network properties from those digests.
- To validate our methodology, we evaluated the digest-graph creation algorithm proposed by Fischer *et al* [4]. We performed extensive simulations of provider-customer and peering relationships between small, medium, and large domains. Our results show that graph-digest based cross-domain fault localization improves speed of inference and protects sensitive network properties while maintaining accuracy in identifying the faults. Our evaluation also reveals specific trade-offs between accuracy and privacy protection warranting further research.
- We have formulated a possible attack against a domain's causal graph. The attack is able to reverse engineer topology information about that domain.

In the next section, we present the required details adopted from previous efforts and develop the terminology used throughout the paper. We continue with a description of our experimental methodology and associated evaluation metrics in Section 3. We then report experimental results for the provider-customer and peering settings, respectively, in Sections 4 and 5. Related work is discussed in Section 6, Finally, we provide some broad interpretations of our results and conclude with a summary of our findings.

THIS PAGE INTENTIONALLY LEFT BLANK

II. FOUNDATIONS

In this section we introduce the foundations of cross-domain fault localization as used in this paper. We introduce the terminology and concepts needed to address the feasibility of cross-domain fault localization.

As defined in [4, 6, 9], a *shared risk group (SRG)* models a component in the network (typically hardware) that could cause a network fault. When the component is malfunctioning, it is said to have *failed*. A *best explanation* for a fault is the set of SRGs whose failure best explains (highest probability) observations of the behavior of the network. The observations are modeled using *observation nodes*. An observation node represents the state of an associated observation (e.g., traffic is flowing over a link). The state of the set of all SRGs and observation nodes is described via a probability distribution modeled with a *network causal graph*. In particular, a network causal graph is a directed acyclic graph capturing a Bayesian model [10] of fault propagation in a communications network, with directed edges between SRGs and observation nodes indicating the existence of a probable causal relationship. Hence, an observation node having state `true` implies that one or more of some set of (connected) SRGs has probably failed.

At a high level, the work of Fischer *et al* applies to arbitrary Bayesian network models however for their examples, they restrict the discussion to the restricted set of models used by the SHRINK [6] fault localization algorithm. SHRINK performs fault localization using Bayesian inference on bipartite causal graphs. In the bipartite model, all edges from an SRG connect only to observation nodes. Each edge has an associated edge weight reflecting the conditional probability of the observation given the SRG failed: $Pr(Observation|SRG)$. The SHRINK algorithm is simple and efficient, serving as an excellent vehicle for evaluating the feasibility of cross-domain fault localization.

In using graph-digest based fault localization, the domain (domain D_j) administrator performing fault localization, receives digests from each of the other n network administrators (domains $D_{i=1 \ldots n}$) involved in the cross-domain fault localization problem. The algorithm used by D_j merges the digests from each of the other domains with its own original (referred to as *undigested*) causal graph and then performs inference to find the most probable set of SRGs explaining the observed faults.

In order to establish a frame of reference for combining local network causal graphs, network domain managers must agree on a set of *shared attributes*. In this work, a shared attribute represents a resource provided by one domain and used in another. It is modeled in the "providing" domain as an observation node and in the "using" domain as an SRG. An example of a shared attribute might be the state of a peering point link between two domains. The providing domain, which may own the physical link, may model the observation state of the link as a shared attribute. The using domain would then model an SRG node, representing resources from the providing domain, as the same shared attribute. The shared attribute nodes are removed during the merge process when combining causal graphs for inference. Care must be taken to ensure that no cycles can form when the causal graphs are joined. In practice, we observe that a simple rule specifying that no shared attribute SRG may have a shared attribute observation node breaks potential cycles in our bipartite model. Some information may be lost and there may be conflict of information challenges. Consider a scenario where the conditional dependencies assigned in one domain do not match the conditional dependencies assigned in another domain. Domain managers must carefully model the events that the shared attributes describe, and agree on how they are implemented.

Fischer *et al* measure inference accuracy by comparing performance of the graph-digest approach against the full disclosure approach. This implementation fundamentally constrains any digest-approach such that it can never do better than full disclosure inference. We take a more direct approach by measuring accuracy of the digest-approach against ground truth failures. By using ground truth as our baseline, we can reason about digest-approach accuracy improvement relative to isolated inference, and the inference cost relative to full disclosure. Next we explore the experimental methodology and metrics used for evaluating graph-digests as a tool for cross-domain fault localization.

THIS PAGE INTENTIONALLY LEFT BLANK

III. EVALUATION METHODOLOGY

This section details the methodology we have developed to evaluate the feasibility of cross-domain fault localization. The crux of our method is a set of metrics for modeling various aspects of the performance of a fault localization design based on the graph-digest approach. The metrics are relatively easy to compute and make it possible to quantitatively and scientifically evaluate the two key questions posted earlier.

In addition, we provide rationales regarding the type of inter-domain relationships and topologies we have chosen to model in our simulation experiments.

A. MODELING INFERENCE GAIN

We address two specific questions regarding the benefits of using a digest based approach:

1. What is the increase in inference accuracy gained by using the digest approach for cross-domain scenarios compared to the accuracy achieved when domains perform inference in isolation?
2. What is the decrease in inference accuracy caused by using the digest approach compared to the accuracy achieved when domains collaborate with full undigested information?

Question 1 above can be paraphrased as "What do we gain by sharing information when troubleshooting a problem?" Question 2 looks at the problem from the other direction: "What do we lose by trying to keep some things secret?" If the answer to Question 1 is "a lot" then the cross-domain fault localization is effective. If the answer to Question 2 is "not a lot" then the graph-digest approach is efficient at realizing the potential accuracy gain of cross-domain fault localization.

1. Accuracy metrics

How is accuracy measured? Consider n domains performing fault localization and let B_T denote the set of actual faults (i.e., the ground truth). Let the best explanation derived by an isolated inference, full disclosure, and privacy preserving approach be B_i for each domain i, B_u, and B_d, respectively. We first consider the case of isolated inference. Clearly if $(B_T - (\cup_{i=1}^{n} B_i)) \neq \varnothing$, then the isolated inference results contain false negatives (some faults weren't found). The hit ratio [4] (i.e. h_s) measures the percentage of correct results in $\cup_{i=1}^{n} B_i$:

$$h_s = \frac{|(\cup_i B_i) \cap B_T|}{|\cup_i B_i|}. \tag{III.1}$$

Likewise if $((\cup_{i=1}^{n} B_i) - B_T) \neq \varnothing$, then the inference results in isolation have false positives. The coverage ratio [4] (denoted: c_s) measures the percentage of faults in B_T that are correctly identified by $\cup_{i=1}^{n} B_i$:

$$c_s = \frac{|(\cup_i B_i) \cap B_T|}{|B_T|}. \tag{III.2}$$

The overall accuracy of isolated inference (denoted: α_s) is the harmonic mean of h_s and c_s:

$$\alpha_s = \begin{cases} 0 & \text{if } h_s = c_s = 0 \\ \frac{2 \cdot h_s \cdot c_s}{h_s + c_s} & \text{otherwise} \end{cases} \tag{III.3}$$

The value of α_s ranges from 0 (zero accuracy) to 1 (perfect inference). Intuitively, a small α_s value indicates a need for cross-domain coordination.

We follow an identical method for the full disclosure and privacy preserving approaches. The accuracy using full disclosure (*undigested* graphs) is calculated:

$$\alpha_u = \begin{cases} 0 & \text{if } h_u = c_u = 0 \\ \frac{2 \cdot h_u \cdot c_u}{h_u + c_u} & \text{otherwise} \end{cases} \tag{III.4}$$

where

$$h_u = \frac{|B_u \cap B_T|}{|B_u|}, \text{and } c_u = \frac{|B_u \cap B_T|}{|B_T|}. \tag{III.5}$$

and the accuracy of the privacy preserving approach is calculated:

$$\alpha_d = \begin{cases} 0 & \text{if } h_d = c_d = 0 \\ \frac{2 \cdot h_d \cdot c_d}{h_d + c_d} & \text{otherwise} \end{cases} \tag{III.6}$$

where

$$h_d = \frac{|B_d \cap B_T|}{|B_d|}, \text{and } c_d = \frac{|B_d \cap B_T|}{|B_T|}. \tag{III.7}$$

Without special consideration, a failure hypothesis involving $x > 1$ indistinguishable SRGs will result in adding x SRGs to the best explanation every time, adversely impacting the hit ratio of the guess. We combine these nodes into a single SRG to calculate the α_u, α_d, and α_s scores. Consolidating indistinguishable SRGs is consistant with the SCORE [8] fault localization algorithm.

To quantify the inference gain from using the privacy preserving approach (i.e., to answer question 1 above), we propose to compute the difference between its inference accuracy and the accuracy achieved by domains in isolation:

$$A = \alpha_d - \alpha_s \tag{III.8}$$

The value of A ranges from -1.0 to 1.0. A positive score means that sharing digests improved fault localization, a score of 0.0 means there was no improvement, and a negative value means that using a graph-digest approach was worse than isolated inference. For example, suppose $B_T = \{S1, S4\}$, $\cup_i B_i = \{S2, S5\}$, and $B_d = \{S1, S5\}$. We have $h_s = c_s = 0$ and $h_d = c_d = 0.5$. Thus, the inference gain A equals 0.5 for this case.

Similarly, we propose to measure the cost of privacy protection (i.e., to answer question 2 above) with the following metric:

$$C = \alpha_u - \alpha_d \tag{III.9}$$

where C should range from 0.0 to 1.0, with a larger value indicating a higher cost. Continuing with the example above, C would be 0.5 if the full disclosure approach achieves perfect accuracy, i.e., $B_u = B_T$, which implies $\alpha_u = 1.0$.

Note that, since the shared models are smaller, the digest based approach require dramatically less computation as compared to the full disclosure approach. In other words, the digest based approach is much more scalable. Section 3 discusses how to quantify this benefit.

B. MODELING PRACTICALITY ISSUES

With respect to the practicality of using graph-digest based fault localization, we ask:

1. How well is privacy protected for realistic metrics?
2. Are the inference gains consistent across different network domain types and sizes?
3. Is the digest based approach scalable in terms of inference running time?

We address these questions below.

1. Privacy metrics

Prior work [4] advocates using the information theoretical Kullback-Leibler (KL) distance [3] to measure how well the digest based approach protects a domain-local property. Although the KL distance is an ideal metric to measure privacy preservation, it is extremely difficult to apply in practice as to compute KL distance one needs to know (or estimate) the prior probability distribution the adversary uses (explicitly or implicitly) to guess the secret. In this paper we explore a more pragmatic approach: characterize the effectiveness of various attacks using a digest to learn specific sensitive properties about the digest's source domain. Specifically, we provide a systematic method for experimentally evaluating attacks against a causal graph.

Modeling a Causal Graph Attack. We emphasize that the focus of this paper is on developing a general evaluation methodology, not developing the most effective attacks on causal graphs. Not surprisingly, our literature search did not uncover previous work addressing attacks on a causal graph. We explored different

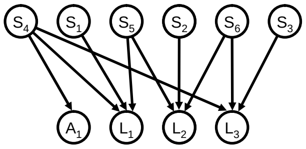

Figure 1. Example causal graph.

properties that a causal graph could reveal about a network, and did not find a meaningful sensitive property for which we could realistically construct a distribution about the property from an adversary' perspective. Instead, we developed a heuristic to learn a domain's topology (routers, switches, physical and VPN links, etc.) from a causal graph of the network.

We implemented our heuristic to specifically attack SHRINK-style bipartite causal graph-digests. We assume that information such as prior probabilities and conditional probabilities have been anonymized by setting all respective values to the same strength. For brevity we present a simplified description of our attack heuristic.

Our attack iterates through all observation nodes in the given graph. If an observation node is a shared attribute (e.g., circuits in provider-customer relationship), we assign all parent nodes to that observation node as gateway routers. Otherwise, we check if the observation node is the child of a shared attribute, and if so, designate all other parents of the observation node as gateway routers. Finally, if neither of the above conditions is met, we find the first SRG parent node with a degree of one. There are exactly three parents to check in this loop of the algorithm since the observation node represents an IP link between adjacent routers and/or switches: one point-to-point link and two routers and/or switches. This SRG can only be a point-to-point link or stub router. The SRG is assigned as a point-to-point link and the remaining two parents are assigned as adjacent routers.

Consider the example causal graph in Fig. 1. The causal graph has SRG nodes $S_1 \dots S_6$, observation nodes $L_1 \dots L_3$, and a shared attribute observation node A_1. Suppose we know that A_1 represents a peering-point shared attribute. We now

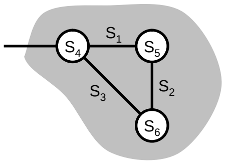

Figure 2. Topology produced by our attack heuristic.

conclude that S_4 is a gateway router. Next we observe that L_1 has 3 parent SRG nodes, with S_1 having a cardinality of 1. We determine that S_1 is most likely a point-to-point link connecting the gateway S_4 and an adjacent router: S_5. We now have nodes S_4 and S_5, and edge (S_4, S_5) in a graph representing the topology. Applying the same reasoning with L_2 and L_3 allows us to add node S_6 and edges (S_5, S_6) and (S_4, S_6), resulting in the topology shown in Fig. 2.

Modeling Attack Effectiveness. There are many properties that a network domain administrator may consider sensitive. However, relatively few of these can be inferred from a causal graph of the domain. Properties such as detailed customer information or operating system details, are most likely abstracted away in a causal graph. We have identified four sensitive properties for evaluation of privacy protection:

- Domain network diameter
- Number of routers in a domain
- Degree of the node with the highest degree in a domain
- Internal reachability between a pair of visible gateways.

For each digest, we first run our attack heuristic on it to infer the domain topology and then derive values for each of the sensitive properties from the topology. Eventually, we obtain a collection of sample values for each property from each target set of scenarios. We use the following metrics to model the effectiveness of the attack against a property.

- Root mean square error ($rMSE$).

  Let $X = \{x_1, x_2, ..., x_m\}$ represent the collection of samples for a set of m scenarios where the property has a fixed true value of P. $rMSE$ for that scenario set is defined as follows:

  $$rMSE = \sqrt{E((X-P)^2)} = \sqrt{\sum_{i=1}^{m}(x_i - P)^2/m} \tag{III.10}$$

  The interpretation of $rMSE$ is straightforward: if the $rMSE$ value is large relative to the true value P, we consider the attack unsuccessful.

- Generalized standard deviation ($gSTD$).

  Usually the standard deviation, like $rMSE$, should be defined with respect to a set of scenarios where the property's true value is fixed. We generalize the definition to consider samples from all scenarios used in the evaluation. Let $\{x_1, x_2, ..., x_M\}$ represent the collection of samples for all M scenarios. $gSTD$ is computed like a usual standard deviation:

  $$gSTD = \sqrt{E((X - E(X))^2)} \tag{III.11}$$

  $gSTD$ has a desirable feature: it captures how well the attack algorithm tracks the fluctuation in the true value of the property. We will articulate more on this point in the Discussion section. The attack is not effective if $gSTD$ is small relative to the sample mean $E(X)$. For this reason, $gSTD$ can be viewed as a good indicator of the KL distance.

2. Generality

Several factors warrant consideration when evaluating the generality of the digest based approach to cross-domain fault localization. First, sizes of real networks may vary greatly. Second, the networks may have diverse topological structures (ring, star, mesh, number of neighbors, etc.). Third, domains may use different inference tools for troubleshooting. Fourth, the privacy and security requirements can differ greatly from domain to domain. Last but not least, a domain may have either a provider-customer or a peering relationship with each neighboring domain.

It is unrealistic to require a digest creation algorithm to work optimally with respect to all of these factors. In this paper, we restrict our evaluation to consider two factors: network size and the type of relationship between domains. We conduct

```
createBipartiteDigest($G$)
Add node $L_{new}$ to $G$
for all $SRG\ S_i \in G$
    if (for all edges $(S_i, L_j) \in G$, $L_j$ is up)
        then Prune $S_i$ and its edges $(S_i, L_j)$
    else
        Collect edges $(S_i, L_j) \in G$ such that $L_j$ is $up$
        if At least one such edge exists
            Add edge $(S_i, L_{new})$
        Prune collected edges $(S_i, L_j)$
Remove all isolated observation nodes $L_i$
for all $SRG\ S_x, S_y \in G$
    if $S_x$ and $S_y$ are indistinguishable
        Aggregate $S_x$ and $S_y$ into $S'_x$ such that $S'_x = S_x \cup S_y$
Rename all SRGs that are not shared attributes
Rename all Observation nodes other than $L_{new}$
```

Figure 3. Algorithm for computing a digest from a bipartite causal graph G. [4]

two sets of experiments for a pair of cross-domain relationship styles: one modeling the provider-customer relationship and the other modeling a peering setting. Note that most, if not all, inter-domain relationships fall into one of these two categories. We construct three topologies (small, medium, and large) for each relationship style to capture a range of domain sizes and to assess the effects of scale.

As our target of evaluation, we have chosen the same digest creation algorithm used by Fischer *et al* [4] with one modification. The algorithm uses simple techniques, such as node and edge pruning, partial evaluation, aggregation, and node renaming. The original algorithm uses Noisy-OR computation to adjust edge strengths in the digest causal graph based on the number of "up" neighbors for a network device. We found that to be very revealing about a domain network topology and, therefore, have replaced the Noisy-OR with logical OR in our implementation of the algorithm. A full specification of the algorithm is depicted in Fig. 3.

Our selection of this algorithm was driven by two considerations: First, it is the only one that is available. The focus of this work is on the evaluation methodology and hence we leave the development of potentially more effective digest creation algorithms to future work. Second, the use of a straw-man algorithm could potentially strengthen the conclusion drawn from the evaluation. If a rather simplistic digest creation algorithm were to perform promisingly, it would be reasonable to conclude positively about the feasibility of cross-domain fault localization when more polished techniques are used.

3. Scalability

Although the SHRINK algorithm achieves polynomial time inference by assuming no more than 3 concurrent SRG failures [6], the algorithm still must consider $\binom{n}{1}+\binom{n}{2}+\binom{n}{3}$ hypotheses [1], with n here denoting the total number of SRGs. The computation complexity for SHRINK is $O(n^4)$. Clearly, by compressing causal graphs, the digest approach will reduce n resulting in a far fewer hypotheses to consider vs. full disclosure. As a result, the digest approach is more scalable in terms of inference running time.

To validate this claim, we propose to directly measure the inference running times of the two approaches. Let t_u and t_d represent the recorded average running times for the full-disclosure and privacy preserving approaches, respectively. We introduce the following metric to quantify the scalability improvement:

$$E = log_{10}(\frac{t_u}{t_d}) \tag{III.12}$$

E measures the order of magnitude of reduction in inference time gained by using the digest approach as compared to full disclosure. A value for E much greater than 0 reflects significant savings in inference time by using the digest strategy, while a value close to 0 reflects little or no savings, and a value less than 0 means that the digest strategy has actually caused an increase in inference time.

[1] After abstracting away the null hypothesis and the "not in the model" hypothesis

C. FAILURE SCENARIOS

There are total of 6 different topologies: We evaluate two types of relationship between domains and for each relationship type we create three topologies (small, medium, and large). For each topology, we collected data for *single* and *double* failure scenarios. If an observation node could exist in another domain that provides evidence about an SRG, we define that SRG as a *cross-domain SRG*. We generated failure scenarios randomly, but to favor scenarios that may require cross-domain fault localization, we constrained failure selection such that at least one failure must be a cross-domain SRG. We evaluated all single failure scenarios that satisfy this constraint. We executed three data collection cycles of fifty failure scenarios each for the double failure scenarios, yielding 150 distinct double failure scenarios for the small and medium topologies. For both of the large topologies, we executed two collection cycles of twenty-five failure scenarios, resulting in fifty distinct double failure scenarios.

As explained earlier we use the SHRINK [6] algorithm to perform inference for each failure scenario. The case of isolated inference is straightforward. Inference was first performed separately on each domain's own causal graph, and the resulting best explanations were then combined into $B_1 \cup B_2$. In the case of full-disclosure, the causal graphs of the two domains were simply combined and the inference was perform on this combined graph to produce B_u. Finally, in the case of privacy-preserving, the causal graph of domain 2 was first processed with the digest creation algorithm. The resulting digest was then combined with the causal graph of domain 1 and inference was performed on the combined graph to produce B_d.

In the next two sections we apply the evaluation criteria discussed above to evaluate the digest strategy against these customer-provider and peering scenarios.

IV. PROVIDER-CUSTOMER SETTING

In this section we employ the above methodology to explore cross-domain fault localization performance in provider-customer situations. In a provider-customer relationship, one domain (the provider) provisions network backbone connectivity to a second domain (the customer). In many cases, the provider's physical topology (e.g., SONET connections multiplexed on fiber) isn't observable by the customer. The customer only sees IP connections entering the edge device on one side of the provider's cloud and exiting on the other. Sometimes only a core router is visible. In any case, many sources of faults aren't visible to the customer. Furthermore, configuration problems on either the customer's or the provider's side may result in faults that aren't readily observable by both parties.

For the provider-customer topologies, with our assumption that failures are not total in the provider network and individual IP flows are not instrumented for fault detection, we deny observations about customer flows to the provider.

A. PHYSICAL TOPOLOGY

We evaluated the digest-based information sharing approach in a simulated provider-customer network setting in which a customer transits the provider domain

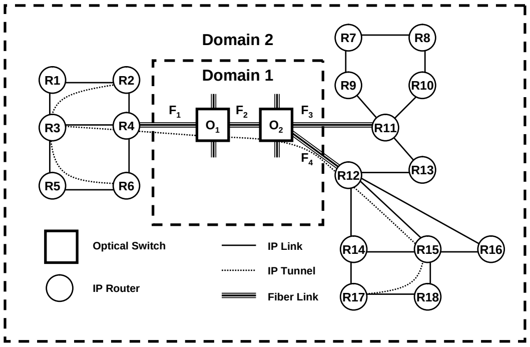

Figure 1. Provider-Customer Physical Topology.

using three leased circuits. We use the topology presented by Fischer *et al* [4] for our small topology depicted in Fig. 1. We grew the customer topology on each side, adding sub-components to reflect realistic network topology elements to create the medium and large network topologies. The small, medium, and large customer network domains have 18, 54, and 204 routers respectively. The sub-components in the expanded network have varying properties, such as node degree and distance between elements in the domain, and are connected in mesh, star, ring, and *ad hoc* topologies. The provider network (Domain 1 in Fig. 1) consists of Optical Digital Cross Connect switches and fiber to transit customer traffic. We focus our study on finding cross-domain faults that occur between the provider domain and one of its customers: Domain 2 in Fig. 1.

B. CAUSAL GRAPHS

As illustrated in Fig. 1, the Domain 1 circuits have 2 optical cross connect switches (O_1 and O_2) and 4 fiber spans ($F_1 \ldots F_4$) as SRGs. In Domain 2 (the customer domain) we model each router ($R_1 \ldots R_{18}$) and point-to-point link between adjacent routers (e.g., $R_1 - R_3$) as an SRG. Every SRG failure in the customer domain generates observations about the failure. We model the IP connections between each pair of adjacent routers; the 3 internal VPN tunnels ($R2 - R3$), ($R3 - R6$), and ($R15 - R17$); the cross-domain IP connections ($R4 - R11$) and ($R11 - R12$); and the cross-domain VPN tunnel ($R3 - R15$) as observation nodes in Domain 2. The three leased circuits underlying the cross-domain IP links serve as the shared attributes for this setting, with Domain 1 modeling the shared attributes as observation nodes, and Domain 2 modeling the corresponding shared attributes as SRG nodes. We have nine cross-domain SRGs from both domains (O_1, O_2, $F_1 \ldots F_4$, R_4, R_{11}, and R_{12}) in the customer-provider setting.

We show the Domain 1 causal graph in Fig. 2. We label the fiber links between the optical switches as $F_{1\ldots4}$, the optical switches as O_1 and O_2, and the shared

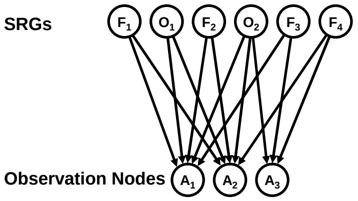

Figure 2. Domain 1 partial causal graph.

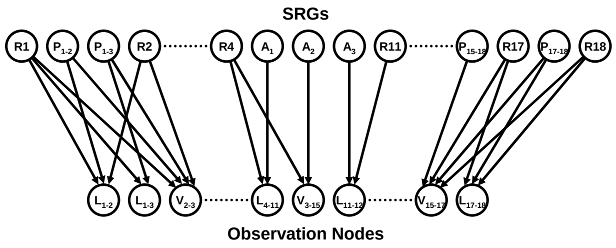

Figure 3. Domain 2 causal graph.

attributes as $A_{1\ldots3}$. Each shared attribute $A_{1\ldots3}$ represents one of the circuits leased by Domain 2.

As the provider domain has many cross-domain SRGs for many customers, we chose the provider domain to perform inference on behalf of its customers for the privacy preserving approach. For each of our failure scenarios, the customer domain generated a digest for inference by the provider domain. We present the Domain 2 small topology causal graph in Fig. 3. We identified routers $R1 \ldots R18$, the point-to-point links P_{x-y} where x and y are the pair of adjacent routers Rx and Ry, and the shared attributes $A_{1\ldots3}$ as the SRGs. The observation nodes are the IP links between the routers $L_{x,y}$ and VPN tunnels $V_{x,y}$, where x and y designate the routers on either end of the connection as in the point-to-point links.

We depict the Domain 2 digest created after observing connection failures L_{4-11} and L_{11-12} in Fig. 2. The SRGs R_4, R_{11}, and R_{12} have been anonymized as $S_{1\ldots3}$, and IP links L_{4-11} and L_{11-12} as L_1 and L_2. Only the special observation node L_{up} observes an “up” state and all other observation nodes report a “down” state. All SRG prior probabilities are set to a uniform value; likewise all conditional dependencies (the edges) have a uniform value.

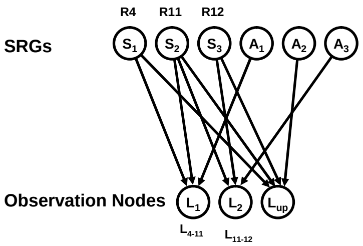

Figure 4. Domain 2 digest.

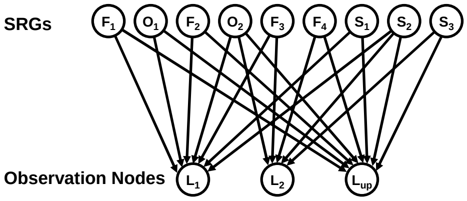

Figure 5. Union.

We use a node collapsing methodology to form a union between the causal graphs, which starts by merging the shared attributes from each causal graph. Next, each observation node inherits all conditional dependencies from all shared attributes on which the observation node is dependent (e.g., if an edge exists from a shared attribute to an observation node, then all edges into the shared attribute from an SRG are copied to that observation node). Finally, the shared attribute nodes are removed. As an example F_1 has an edge to A_1 in the Domain 1 causal graph (Fig. 2) and A_1 has an edge to L_1 in the Domain 2 digest(Fig. 3), thus F_1 gains an edge to L_1 in the causal graph union. The union of the Domain 1 causal graph with the Domain 2 digest is depicted in Fig. 5. For the interested reader, SHRINK inference returns F_3 as the best explanation in both the full disclosure and privacy preserving approaches for this sample scenario.

In our topology (Fig. 1) F_1, F_2, and O_1 are indistinguishable to SHRINK, and fully $\frac{1}{3}$ of our failure scenarios contains one of these three SRGs as having failed. We combine these nodes into a single SRG to calculate the α_u, α_d, and α_s scores.

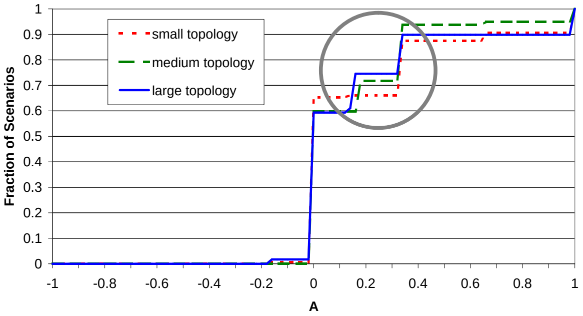

Figure 6. CDF of A metric for the 3 topologies.

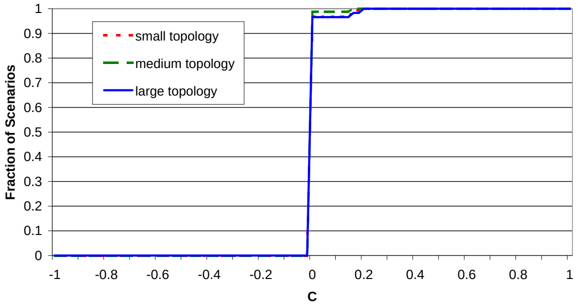

Figure 7. CDF of C metric for the 3 topologies.

C. EVALUATION OF ACCURACY RESULTS

For all but 2 of 377 tested scenarios $\alpha_d \geq \alpha_s$, resulting in non-negative accuracy improvement scores A (Fig. 6). The average A score is 0.186, 0.152, and 0.175 for the small, medium, and large topology respectively. The maximum A score for each topology is 1.0. Although A can be negative and worsen the inference results, we observe an accuracy improvement in more than 28% of our test scenarios for each topology (indicated by a circle in Fig. 6). Our results indicate that scaling the domain size has little impact on the accuracy of B_d, $B_1 \cup B_2$, or A with respect to B_T.

The cost metric C depicted in Fig.7 shows a minimal cost in using the digest approach. The cost to inference equals zero in all but nine test cases, meaning that the digest approach achieved the same accuracy as the full disclosure approach in 97.6% of the 377 tested scenarios. The C score average is 0.005, 0.002, and 0.006 and maximum value is 0.2, 0.17, and 0.2 for the small, medium, and large topology c

	Small	Medium	Large
Degree	1.73 (4)	4.49 (7)	5.61 (8)
Diameter	2.13 (5)	8.41 (11)	20.73 (23)
Routers	9.59 (18)	45.18 (54)	195.88 (204)

Table 1. Privacy metric rMSE versus (true value).

	Node Degree	Domain Diameter	Number Routers
gSTD	0.97	0.74	2.54
E(X)	2.58	2.74	8.72

Table 2. Privacy metric gSTD versus sample mean.

respectively.

The digest algorithm in Fig. 3 potentially degrades A. The logical-OR treatment for edges to L_{up} removes evidence about the liveness of an SRG. Additionally, the aggregation step, exaggerated by using uniform prior probabilities, lumps additional SRGs into a best explanation for α_d. Since all equipment identified in a hypothesis would have to be checked, we unravel all SRGs that have been aggregated into an SRG in a best explanation. Consequently, aggregation potentially adversely affects h_d, and ultimately α_d and A. In spite of the information loss, the privacy preserving approach performed remarkably well as discussed above.

D. PRIVACY EVALUATION RESULTS

To compute the privacy protection for the customer we attacked each digest using our heuristic as described in Section 3. We did not attempt to hide information and did not perform post-processing of the digests to reduce the information leaked, but rather tested to see how much information leaked using the simple digest algorithm described in Fig. 3.

We can see in table 1 that the generalized standard deviation for each privacy

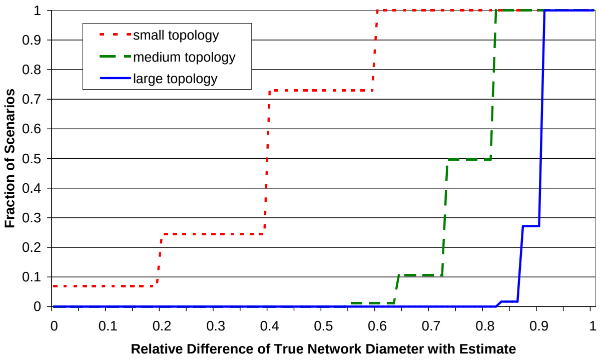

Figure 8. CDF relative error for the diameter property.

metric is low compared to the mean. This result means that there is little variation in the amount of information learned about each sensitive property from each digest attack. As depicted in Table 2, the root mean squared error is high relative to the true value. This outcome means that the information learned from the attacks is generally far from the true values. These results suggest a reasonable level of privacy protection considering our use of a straw man digest creation algorithm for our attack heuristic.

To provide more detail, we calculated the relative error of the attack results versus the true value for each sensitive property. We present the results expressed with a CDF (less reachability) for each domain size in Figs. 8 - 10. We discuss the reachability results next.

The reachability metric is binary with a one, the true value, representing internal reachability between two externally visible gateways in the customer domain. These gateways are two hops distant in the customer domain for each topology. Only 7 of the 377 failure scenarios identified the reachability between the nodes.

As the network size increases, the inferred diameter deviates further from the true value (Fig. 8). At 50% mass of our experiments, the relative distance is approximately 40%, 80%, and 90% for the small, medium, and large topologies respectively. This result bodes well for the inherent protection provided by the digest approach as a network domain size scales up.

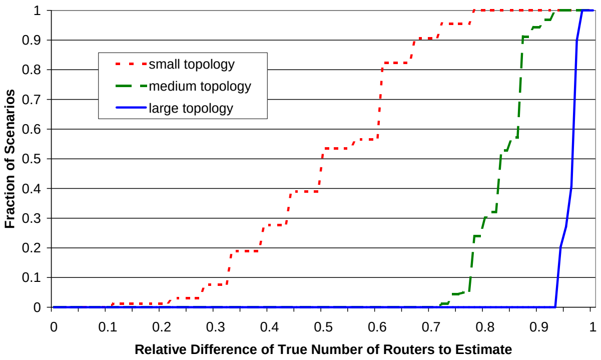

Figure 9. CDF relative error for the number of routers property.

As shown in Fig. 9, we found the relative distance between the number of routers in a network and the number detected from attacking the digest to increase with topology size as with the diameter. Intuitively this results makes sense as each attack discovers one or more anonymized subgraphs that are either connected to one or more gateways or represent anonymized internal topology. In general the topology learned consists of a neighborhood around one or two routers. Multiple failures whose neighborhoods intersect allow a larger portion of the topology to be inferred. Each digest only provides small snapshot of the Domain 2 causal graph, largely explaining the low gSTD results.

When a failure impacts a VPN tunnel, as do $\frac{7}{9}$ of the failure scenarios, information about the neighborhood around each router on the tunnel is revealed. The VPN tunnels do have an inherent protection feature in that an IP link representing the VPN tunnel will most likely have more than three parents in a digest. This creates ambiguity in determining router adjacencies along the tunnel for the attack heuristic we used. This ambiguity causes some minor variation in the gSTD results.

We seeded the topologies with an unfavorable setting for the node degree sensitive property by placing a router with high degree at the gateway in the customer domain. The node aggregation and Noisy-OR steps performed by the digest algorithm (Fig. 3) did surprisingly well in hiding the true value of the node degree for some of the attacks (Fig. 10). The true degree was revealed in approximately 13% of the

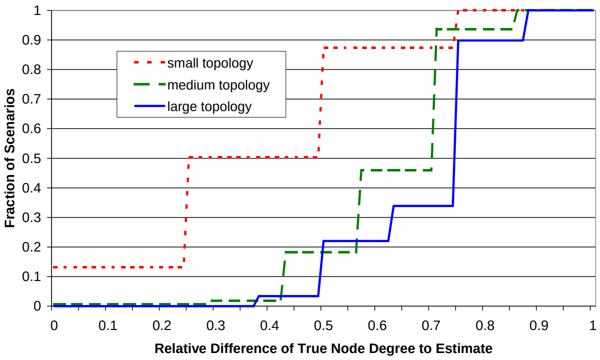

Figure 10. CDF relative error for the node degree property.

	Small Topology	Medium Topology	Large Topology
E	1.00	2.81	4.84

Table 3. Scalability results.

attacks, and the property did not scale with the network domain size. We would expect better inherent protection if we had placed the high degree node in interior of the Domain 2 topology.

From the privacy results we suggest that a privacy metric whose true value naturally grows with the sheer size of a domain receives inherent protection using a digest approach as the size of a network domain scales up. The network diameter and the number of routers naturally grow with a network domain's size, while a high degree node or an interior path between two gateways remains fairly static: an attack either finds it, or it doesn't.

E. SCALABILITY EVALUATION RESULTS

To compute scalability E we measured the average elapsed real time to compute SHRINK results for up to three failures for the small, medium, and large topologies respectively. The simulations were run on a 1.61 GHz computer with 960 MB RAM running Windows XP, service pack 3.

As expected, the SHRINK running time increased significantly as the number of SRGs increased. We can see the increase in scalability E by using the privacy preserving approach in Table 3. Of particular note, inference time improved from hours to mere minutes on the large topology.

V. PEERING DOMAINS

In this section we examine fault localization performance of two peering network domains. The two domains share multiple peering points and web service connections. Ownership of the shared links and hosting of the services is equally distributed between the two domains. IP link and web service failures are fully visible, and we consider device failures as total: an SRG failure causes an observable failure event.

A. PHYSICAL TOPOLOGY AND CAUSAL GRAPHS

As with the provider-customer relationship, we start with the topology presented by Fischer *et.al.* [4], and modify and grow our topology size to incorporate realistic network domain subcomponents. We present our small physical topology in Fig. 1. The peering domains in the small topology have two peering points $(R_4 - R_{11})$ and $(R_6 - R_{17})$ and four cross-domain server-server connections. The servers are depicted as $W1 \dots W5$ (we do not depict the connections for brevity). We model the medium topology with four peering points and eight web service connections, and the large topology with eight peering points and sixteen web service connections.

We model the SRG and observation nodes as in the customer-provider rela-

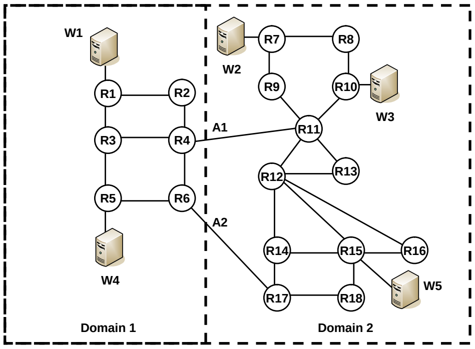

Figure 1. Peering Domains Physical Topology.

tionship, and use the same notation. The set of cross-domain SRGs from which each failure scenario must have failed component contains every peering point router and link, and every router and link on the shortest path between the servers for each web service. We identified 24, 42, and 68 cross-domain SRGs in the small, medium, and large topologies respectively.

We have two types of shared attributes for this scenario. The first type represents the peering links between the domains. For each link, we chose one domain to own the link, and this domain models its observation node as a shared attribute. The peering domain models the corresponding shared attribute as an SRG node. Both domains see the state of the IP link between the domains. The second type of shared attribute describes whether a pair of servers can connect with each other. For each web service, the domain hosting the service models the observation node for the connectivity with a shared attribute, and the domain on the client side models the corresponding shared attribute as an SRG.

B. ACCURACY EVALUATION RESULTS

Initially, we had puzzling inference results as the α_s, α_u, and α_d scores were all low. SHRINK [6] tends to omit point-to-point links and stub routers from multiple failure scenarios using the default settings, instead of attributing the evidence of failure about these components to an error in the SRG database. Although merely a nuisance in the provider-customer setting, the problem became magnified in the peering domain setting due to the large number of links identified as cross-domain SRGs (e.g. the peering links and links on the web service shortest paths). Additionally, failures with low probability mass in one domain caused ambiguous inference results for B_i in the other domain. Our SHRINK model did not include a method for the inference to return $B_i = \varnothing$, which became a necessary feature in the peering domain setting.

To counter the issue of SRG omission, we lowered the prior probabilities of the

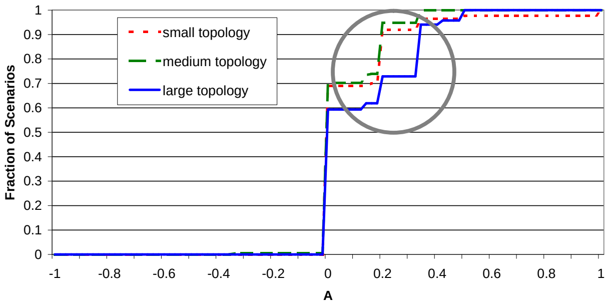

Figure 2. CDF of A metric for the 3 topologies.

SRG nodes from 10^{-5} to 10^{-3}. Now the inference engine prefers to add an additional SRG first, and assumes an incorrect SRG database mapping second. To correct the null hypothesis problem, we implemented a low probability "Not I" node which indicates no failures internal to a domain. Using the low probability node is consistent with SHRINK.

The accuracy improvement metric A for the peering topologies is depicted in Fig. 2). In all but 1 of 484 tested cases $\alpha_d \geq \alpha_s$, resulting in non-negative accuracy improvement scores A. In our tested scenarios, we observe a minimum accuracy improvement of more than 26% for each topology (highlighted with a circle in Fig. 2). The A score average is 0.09, 0.062, and 0.124 and maximum value is 1.0, 0.33, and 0.5 for the small, medium, and large topology respectively. Our results indicate that scaling the domain size has little impact on the accuracy of B_d, B_i, or A with respect to B_T. We see a slightly greater improvement in the large topology, which we attribute to the rich number of cross-domain web service connections.

As shown in Fig.3, the cost metric C is 0.0 (no cost) for all failure scenarios. The results mean that for all 484 tested failure scenarios, the digest approach achieves the same inference results as the full disclosure approach.

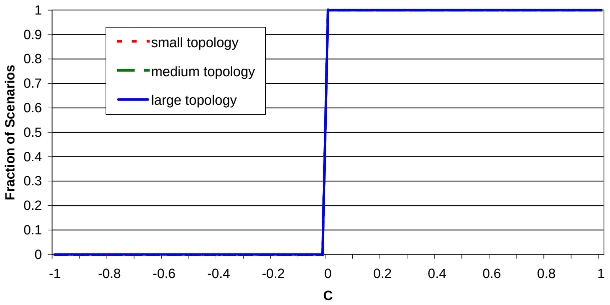

Figure 3. CDF of C metric for the 3 topologies.

	Small	Medium	Large
Degree	2.41 (5)	4.39 (7)	5.50 (8)
Diameter	2.00 (5)	7.33 (10)	22.80 (25)
Routers	4.13 (12)	22.14 (30)	123.65 (130)

Table 1. Privacy metric rMSE versus (true value).

C. PRIVACY EVALUATION RESULTS

We can see in Tables 1 that the generalized standard deviation for each privacy metric is low compared to the mean. As in the provider-customer setting, there is little variation in the amount of information learned about each sensitive property from each digest attack. We show the root mean squared error results in Table 2. Since the rMSE values are high relative to the true value, the information about the sensitive properties learned from the attacks results are generally far from the true values. The results from both the provider-customer and peering settings are encouraging, and a more robust digest creation algorithm can surely improve on the results achieved by our straw man.

	Node Degree	Domain Diameter	Number Routers
gSTD	1.19	0.86	2.73
E(X)	2.76	2.78	7.85

Table 2. Privacy metric gSTD versus sample mean.

Next we provide additional privacy protection data by calculating the relative distance of our attack results against the true values for each sensitive property.

Our attacks never revealed the internal reachability between the visible gateways. We believe that this unexpected result is due to an increased distance of one hop between the gateway routers compared to the provider-customer gateways in our small and medium physical topologies. The tested gateways are three, three, and two hops distant in small, medium, and large topologies respectively.

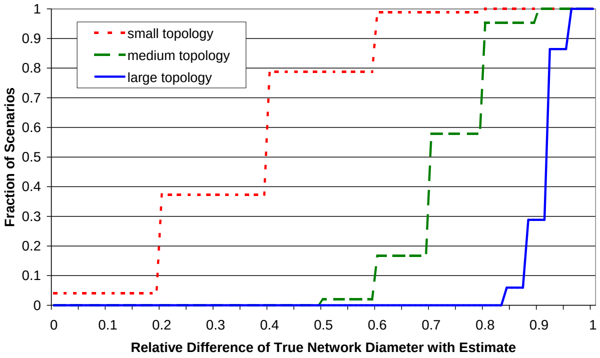

Figure 4. CDF relative error for the diameter property.

The relative distance between the true network diameter and the attack estimate scaled with the topology size as shown in Fig. 4. At 50% mass of our experiments the relative difference was approximately 40%, 70%, and 90% for the small, medium, and large topologies respectively.

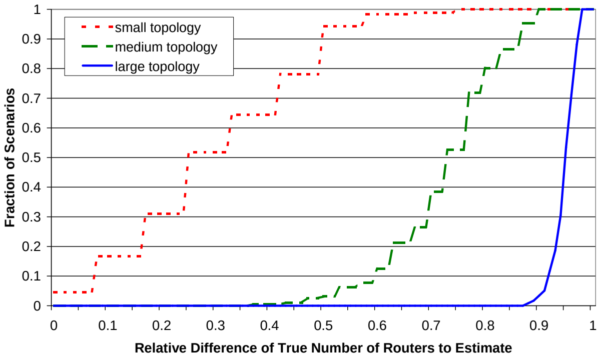

Figure 5. CDF relative error for the number of routers property.

Likewise, protection for the number of routers increased with the size of the topology as seen in Fig. 5. The difference between the true value and attack estimate for 50% mass of the experiments is approximately 25%, 65%, and 95% for the small, medium, and large topologies respectively.

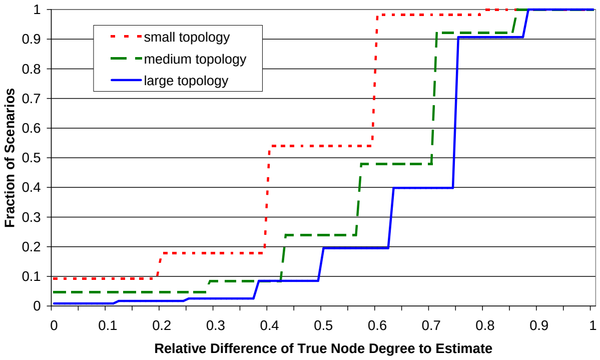

Figure 6. CDF relative error for the node degree property.

Privacy protection for maximum node degree scaled only slightly in the peering domain relationship (Fig. 6) due to several nodes of higher degree in the internal topology of domain D_2. The node aggregation and Noisy-OR steps of the digest creation algorithm contributed to hiding the true value of the highest-degree node for most of the attacks, but approximately 9% of the attacks revealed the true high node degree.

Again we see increased inherent protection for privacy metrics that scale with domain size as the domain scales up. A stronger digest algorithm and post-processing of a digest to remove any information over a predesignated threshold will intuitively strengthen a digest against entropy loss to attack. Additionally, more robust attack procedures would provide greater confidence about the privacy protection of a digest algorithm.

	Small Topology	Medium Topology	Large Topology
E	0.71	1.29	1.72

Table 3. Scalability results.

D. SCALABILITY EVALUATION RESULTS

The scalability (speed) improvement for the peering domain scenario (Table 3), while a significant and encouraging result and achieves order of magnitude improvement, is not as dramatic as that observed in the provider-customer setting. In the peering scenario the domain performing inference has a larger structure, resulting in a greater number of hypotheses for the inference engine to consider.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. RELATED WORK

Steinder and Sethi introduce fault localization as the second step in fault diagnosis following *fault detection* and preceding *testing* [2, 7, 14]. Network administrators use fault localization techniques to discover best hypotheses explaining the observations detected in the fault detection step. Myriad techniques have been developed for intra-domain fault localization, including: rule-based systems, model-based systems, case-based systems, neural networks, decision trees, model traversing techniques, code-based techniques, Bayesian networks, dependency graphs, causal graphs, and phrase structured grammars [14].

The SCORE [8], SHRINK [6], and Sherlock [1] approaches form the state of the art for leveraging causal graphs for fault localization. SCORE uses a set covering approach for finding the best explanation (set of failed SRGs) for observed outages based on a bipartite graph. SHRINK enhances the model to allow probabilistic inference by attaching edge weights that are combined using the noisy-OR [10] model to form conditional probability tables for each observation node. Sherlock further extends these approaches with multilevel causal graphs.

We tested our evaluation methodology by using the cross-domain framework proposed by Fischer *et al* [4]. Steinder and Sethi [15] also proposed a cross-domain fault localization solution. However, it is specifically designed for hierarchically organized networks. This approach locates the source of an end-to-end service failure through distributed coordination between the domains along the path of the failure. In addition to an existing domain hierarchy, the approach relies on full knowledge of each end-to-end data path at the domain level.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. DISCUSSION

In this section, we first provide two broad interpretations of our experimental results. We then briefly discuss some of the limitations of our experiments.

A. BROAD INTERPRETATIONS

A new direction for digest creation. One of the surprises from our experiments is that the proposed metric $gSTD$ is much more effective than we expected at gaging the performance of the digest-creation algorithm in hiding the values of sensitive properties. While further investigations are required to validate the generality of such effectiveness, the results give weight to an alternative approach for creating graph-digests: Instead of allowing a digestion algorithm to produce variable sized digest causal graphs, we may constrain the size and/or structure of the graph digest *a prior*, similar to the way in which a secure hash function has a predefined fixed width in bits (e.g., 512 bits for SHA-512) for all hash values. The primary advantage of this approach is that $gSTD$ would be small regardless of scenarios. However, this approach also brings up a challenge. By restricting the size and structure of a graph digest, it might be difficult to encode within it sufficient information to support inference for large scenarios. We believe this is an interesting and important topic for future work.

Importance of cross-domain issues. The experimental results reaffirm our observation that we need more research efforts in this space. In all topologies simulated, we have discovered a number of scenarios where domains cannot troubleshoot effectively in isolation. We expect a large portion of the real world scenarios to be more complicated than what we evaluated. Therefore, the need for cross-domain solutions is real. In addition to the development of better metrics and algorithms, an emphasis should be placed on the creation of *new theories* for reasoning about what can and cannot be achieved in balancing the trade-off between inference accuracy and privacy protection. Appropriate mechanisms, trust models, and policy must also be developed to support the exchange of causal graph digests and other relevant information (e.g., shared attributes) between domains in collaboration.

B. LIMITATIONS

Beyond visual checks of ”Does this seem reasonable?”, we have not validated how well our simulations model reality. There is little or no publicly available data to allow this validation. Obtaining troubleshooting records from one domain operator is challenging enough. Collecting such sensitive data from a group of connected domains is almost impossible.

We did not model observation errors or missing observations when evaluating inference accuracies. Such events are common in the real-world due to software bugs or misconfigurations. We expect them to have a similar impact to all approaches and, therefore, introduce very small perturbations to the A and C metrics.

We acknowledge some inherent bias in attacking the digests using attack heuristics that we developed. This conflict stands as a necessary evil as our literature search did not uncover a methodology for attacking network causal graphs. We are able to reveal the entire network structure of our undigested causal graphs using our attack heuristics, indicating a sound baseline attack method.

Our core network causal graph model (SHRINK [6]) has a very simple structure. The structure has the advantage of easy inference but lacks expressiveness. In particular, the bipartite nature makes compositing levels difficult. The Sherlock [1] paper gives a more expressive model with much of the inference speed advantages. Expanding the expressive power of the causal graph model requires new algorithms for specifying shared attributes, combining graphs and for constructing digests. The current algorithm for constructing digests incorporates network domain knowledge. Techniques from the artificial intelligence and statistics communities for approximating statistical distributions could be leveraged to produce smaller and more accurate digests. Finally since performing fault localization with digests is significantly faster than without, perhaps digests can be used internally in very large domains to yield faster inference.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. CONCLUSIONS

We have presented the first comprehensive evaluation of the feasibility of cross-domain fault localization. Our evaluation is systematic and complete with regarding to all the proposed performance metrics.

Our goal was to answer the following questions:

1. Does cross-domain fault localization offer the kinds of benefits warranting further research?
2. Can it provide deployable and acceptable privacy protection with manageable complexity?

The answer to both of these questions was a strong “Yes”. Cross-domain fault localization, both with and without digests, performed quite well at finding the faults in all of the scenarios. Of course, in practice not using digests is a non-starter — domain administrators will be simply unwilling to reveal their complete topologies. This brings us to the second question. The digest approaches did provide significant performance gains compared with localization performed in isolation while measurably protecting the sensitive properties we tested. The use of digests dramatically increases the deployability of cross-domain fault localization by decreasing inference time by two to three orders of magnitude. There is still some complexity involved in determining which shared attributes must be modeled, but this effort should be done anyways. Domain administrators need to know what services they are using and providing.

While the answer to both these questions was a strong yes, we did discover several opportunities for further research and enhancement including richer causal graph models and better digest algorithms. This need underscores the importance of having a repeatable evaluation methodology.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Mr. Manuel Hermosilla
   Defense Systems Agency
   701 South Courthouse Road
   Arlington, VA 22204-2199

4. Dr. Raju Namburu
   Army Research Lab
   2800 Powder Mill Road
   Adelphi, MD 20783-1197

5. Jeff Waters
   Space and Naval Warfare Systems Center
   SSC San Diego Code 53621
   53560 Hull Street, San Diego, CA 92152-5001

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] P. Bahl, R. Chandra, A. Greenberg, D. A. Maltz, and M. Zhang. Towards highly reliable enterprise network services via inference of multi-level dependencies. *Proceedings of the 2007 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 13–24, 2007.

[2] M. Y. Chen, A. Accardi, E. Kiciman, J. Lloyd, D. Patterson, A. Fox, and E. Brewer. Path-based faliure and evolution management. *Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation-Volume 1 table of contents*, pages 23–23, 2004.

[3] T. M. Cover, J. A. Thomas, J. Wiley, and W. InterScience. *Elements of Information Theory.* Wiley-Interscience New York, 2006.

[4] W. Fischer, G. Xie, and J. Young. Cross-domain fault localization: A case for a graph digest approach. *Proceedings of Internet Network Management Workshop*, 2008.

[5] X. Huang, S. Zou, W. Wang, and S. Cheng. Mdfm: Multi-domain fault management for internet services. *Management of Multimedia Networks and Services: 8th International Conference on Management of Multimedia Networks and Services, MMNS 2005, Barcelona, Spain, October 24-26, 2005: Proceedings*, 2005.

[6] S. Kandula, D. Katabi, and J. P. Vasseur. Shrink: a tool for failure diagnosis in ip networks. *Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 173–178, 2005.

[7] I. Katzela, A. Bouloutas, and S. Calo. Centralized vs. distributed fault localization. *Proceedings of the fourth international symposium on Integrated network management IV table of contents*, pages 250–261, 1995.

[8] R. R. Kompella, J. Yates, A. Greenberg, and A. C. Snoeren. Ip fault localization via risk modeling. *Proc.Networked Systems Design and Implementation*, 2005.

[9] D. Larrabeiti, R. Romeral, I. Soto, M. Uruena, T. Cinkler, J. Szigeti, and J. Tapolcai. Multi-domain issues of resilience. *Transparent Optical Networks, 2005, Proceedings of 2005 7th International Conference*, 1, 2005.

[10] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, 1988.

[11] P. Reynolds, J. L. Wiener, J. C. Mogul, M. K. Aguilera, and A. Vahdat. Wap5: black-box performance debugging for wide-area systems. *Proceedings of the 15th international conference on World Wide Web*, pages 347–356, 2006.

[12] C. E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949.

[13] M. Steinder and A. Sethi. Probabilistic fault diagnosis in communication systems through incremental hypothesis updating. *Computer Networks*, 45(4):537–562, 2004.

[14] M. Steinder and A. S. Sethi. A survey of fault localization techniques in computer networks. *Science of Computer Programming*, 53(2):165–194, 2004.

[15] M. Steinder and A. S. Sethi. Multidomain diagnosis of end-to-end service failures in hierarchically routed networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(1):126–144, 2008.

[16] D. G. Thaler and C. V. Ravishankar. An architecture for inter-domain troubleshooting. *Journal of Network and Systems Management*, 12(2):155–189, 2004.