# Privacy Principles for Sharing Cyber Security Data

Gina Fisk*, Calvin Ardi†*, Neale Pickett*, John Heidemann†, Mike Fisk*, Christos Papadopoulos‡

*Los Alamos National Laboratory
Los Alamos, New Mexico 87545
{gina, neale, mfisk}@lanl.gov

†USC/Information Sciences Institute
Marina del Rey, California 90292
{calvin, johnh}@isi.edu

‡Colorado State University
Fort Collins, Colorado 80523
christos@colostate.edu

***Abstract*—Sharing cyber security data across organizational boundaries brings both privacy risks in the exposure of personal information and data, and organizational risk in disclosing internal information. These risks occur as information leaks in network traffic or logs, and also in queries made across organizations. They are also complicated by the trade-offs in privacy preservation and utility present in anonymization to manage disclosure. In this paper, we define three principles that guide sharing security information across organizations: Least Disclosure, Qualitative Evaluation, and Forward Progress. We then discuss engineering approaches that apply these principles to a distributed security system. Application of these principles can reduce the risk of data exposure and help manage trust requirements for data sharing, helping to meet our goal of balancing privacy, organizational risk, and the ability to better respond to security with shared information.**

## I. Introduction

Various laws around the world, such as the EU's Data Protection Directive and U.S. Health Insurance Portability and Accountability Act (HIPAA), establish privacy requirements for an individual's data. U.S. policies acknowledge this sensitivity for research in computer security. For example, the National Science Foundation expects researchers to share data, but requires appropriate safeguards to protect the privacy of individuals [1]. While these policies constrain data sharing, there is also a need to share data. For example, in the U.S., the proposed Cyber Intelligence Sharing and Protection Act (CISPA) "directs the federal government to provide for the real-time sharing of actionable, situational cyber threat information between all designated federal cyber operations centers to enable integrated actions to protect, prevent, mitigate, respond to, and recover from cyber incidents" [2]. Since data from cyber incidents often contains personal information from computers and smartphones, systems for sharing cyber security information must consider privacy issues as they exchange and analyzed information.

Solving security research and operational security problems increasingly requires *sharing data across and within organizations*, but it must do so while considering the challenges of individual privacy. Research increasingly emphasizes open data that "anyone can freely access, use, modify, and share for any purpose" [3]. Computer attackers often access systems from multiple organizations to hide their tracks; defenders must unravel these paths to understand attacker command and control systems. In both cases, information sharing is necessary to make progress, but carries significant privacy and security risks. Additionally, laws (such as wiretap laws) and ethical requirements constrain sharing [4], and even collection of such data may raise new risks of data theft [5].

In spite of the risks involved with sharing information across organizations, there is a compelling need to share information to solve problems that are inherently distributed. Distributed Intrusion Detection Systems (IDSes) are one way to correlate security data across large networks [6], [7]. Ontologies like the Vocabulary for Event Recording and Incident Sharing (VERIS) [8] and Structured Threat Information eXpression (STIX) [9] provide a common language for sharing cyber security events. In fact, unstructured sharing is probably the most common method of cross-organizational sharing: invite-only security mailing lists and informal or semi-formal sharing networks are effective today in network operations. However, distributed IDSes are limited in what information can be shared, and unstructured sharing assumes humans in-the-loop to interpret data. Neither of these approaches matches the flexibility and detail needed for automation, with careful mechanisms to manage privacy and information disclosure.

We are currently working on *Retro-Future*, a system that allows controlled information sharing between organizations (and within an organization), along with tools to allow each site to capture and review network information (traffic, routing, naming) relevant to security events. In the process of securing a network, participating organizations will compare anomalies and vulnerabilities with each other to effectively and quickly discover and recover from network attacks. A system such as Retro-Future may be used to exchange external and internal network traffic data with more flexibility and more privacy protection.

The goal of Retro-Future is to design and implement a framework that allows participating organizations to compare anomalies and vulnerabilities with each other to effectively and quickly discover and recover from network attacks, yet balance privacy, organizational risk, and the ability to improve response to security events by controlling disclosure, limiting access privilege, and using anonymization where feasible.

The contribution of this paper is to present principles that can be used to design privacy protection into a system for controlled cyber security sharing across multiple organizations. Our work is guided by three privacy principles (§ II): Least Disclosure, Qualitative Evaluation, and Forward Progress. We then present engineering approaches that can be used to apply these principles in our proposed system (§ III). Our goal is to build on prior work in research ethics [10] and ethical research in Information Technology [4], and we expect to explore these principles as we develop the Retro-Future system for sharing.

## II. Principles and Corollaries

We propose three principles and corresponding corollaries to guide the privacy architecture of a distributed data access system.

This material is based upon work supported by Department of Homeland Security Science and Technology Directorate, Cyber Security Division, via SPAWAR Systems Center Pacific under Contract No. N66001-13-C-3001. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of SSC-Pacific.


DOI 10.1109/SPW.2015.23

**Privacy Principles for Sharing Cyber Security Data**

| *Name* | *Concept* | *Implementation Benefits* | *Consequences For Ignoring* | *Approaches* |
|---|---|---|---|---|
| **Principle of Least Disclosure** | Systems should strive to disclose as little to others as possible, while still sharing. | An organization's risk exposure is reduced because less data is released. | Without sharing, common security issues go unresolved. Secondary Privacy Damage. | Minimal Requisite Fidelity, Moderated Queries, Poker Queries, Anonymization |
| *Corollary 1: Internal Disclosure* | Collecting data, even if it has not been released, is a source of potential disclosure. | Protecting information before it is released will reduce inadvertent disclosure | Inadvertent disclosure of unreleased data. | Data Confinement, Secure Data Archive, Data Aging |
| *Corollary 2: Privacy Balance* | One must balance disclosure by the querier and the responder. | The privacy needs of the organization and the individual will be balanced. | Privacy Diffusion and Secondary Privacy Damage | Controlled Disclosure, Organizational Sharing Policy |
| *Corollary 3: Inquiry-Specific Release* | Access to data should be moderated and limited. | Queries minimize secondary privacy damage. | Privacy Diffusion and Secondary Privacy Damage | Least Privilege, Moderated Queries, Poker Queries |
| **Principle of Qualitative Evaluation** | One must balance (subjectively) costs and benefits for privacy and progress. | Prevents acceptance of privacy algorithms as complete solutions. | Organizations may encounter unexpected roadblocks. | IRBs, Separate mechanism from policy |
| *Corollary 1: Legal Constraints* | Organizations must live within to legal and ethical constraints. | It becomes easier to adapt to new laws and constraints. | Research and sharing may be shut down if laws are ignored. | IRBs, Separate mechanism from policy |
| *Corollary 2: Technical Limitations* | Technical methods alone are not a viable approach to privacy. | A combination of technical and policy methods can improve risk management. | Organizations may be blindsided by the limitations of specific algorithms or techniques. | IRBs, Separate mechanism from policy |
| **Principle of Forward Progress** | Organizations must not become paralyzed by Least Disclosure and Qualitative Evaluation. | Sharing is allowed, albeit in a controlled manner with consideration of benefits. | Organizations may become paralyzed by laws and restrictions. | Controlled Disclosure |

TABLE I. SUMMARY OF PRINCIPLES, COROLLARIES, AND APPROACHES

**Principle of Least Disclosure:** We see *least disclosure* as an overarching principle to privacy architecture. In a distributed data access system, information is disclosed by both the query to and response from the system. Anonymized replies and queries mitigate some of this problem, but information leaks through de-anonymization [11]–[13] gives data owners pause when considering sharing. Although any release of information adds a potential risk, the guidelines of Least Disclosure encourage minimization of shared information to minimize such risk.

Vast datasets can be used in a variety of ways, from analyzing the entire dataset to small subsets of the data. Rather than making complete datasets available, by applying the Principle of Least Disclosure, a user is given only a subset of the data constrained by its owner's assessment of benefits against risk. In a sense, this principle extends the requirement of "purpose limitation" from European privacy law: not only should data have a specific goal in mind, but additionally *what* is shared or stored should be minimized.

There are three associated corollaries to this principle. First, *Internal Disclosure* recognizes that risk and compartmentalization are not only properties of an organization, but often must be considered across groups inside the enterprise as well. Second, *Privacy Balance* is the idea that greater disclosure by the querier means less disclosure by the responder; this tradeoff should be consciously chosen. This balance is illustrated in (§ III-C). Lastly, the corollary of *Inquiry-Specific Release* is the concept that access to information should be given based on approved specific uses and then providing access to the minimal amount of information that must be disclosed for that usage through a query interface.

**Principle of Qualitative Evaluation:** We recognize that there are both *legal* and *technical* constraints to sharing information, and neither alone is sufficient. One must weigh subjective decisions to manage risks in sharing, and these processes are unlikely to be realizable with only computer algorithms. We further reject the assumption that a single privacy metric, algorithm, or new mechanism will "solve" the problem of protecting privacy while enabling meaningful sharing. Quantitative and objective risk analysis is insufficient, and systems must have mechanisms to enforce any potential policy decision that may be made on the basis of subjective legal and ethical review. This process should include end-user privacy concerns and institutional requirements.

There are two associated corollaries with this principle. The first is that of *Legal Constraints*. The system must operate within legal and ethical constraints. Increasingly, best practices subject IT research to Institutional Review Board review (as suggested in the Menlo report [4]). Similarly, data sharing must be flexible enough to accommodate and support ethical and legal constructs and their subjective determinations. Organizations should strive to *separate mechanism from policy* so that subjective decisions can be implemented quickly. Organizations cannot rely solely on quantitative measures, differential privacy, or provable privacy to meet these constraints.

The second corollary is that of *Technical Limitations*. Organizations must accept that technical methods alone cannot insure complete privacy while allowing forward progress. We see a *combination* of technical and policy methods as necessary to provide sharing in the face of risk.

**Principle of Forward Progress:** We believe that sharing data among participating organizations is necessary to improve security and promote progress in research and understanding. Zero tolerance for disclosure prevents the legal, ethical, and constructive use of data. Information sharing needs to occur in some mutually agreeable fashion. While sharing nothing aside from what is legally required might seem like the safest and least risky move, a lack of openness and willingness to share will prevent progress into developing new techniques for study and the exchange of information beneficial to all. From a game-theoretic perspective, the outcome of sharing should be a positive sum.

## III. ENGINEERING APPROACHES

While these principles guide our thinking, they are intentionally high-level. We next consider how they apply to the practical matter of building a distributed information sharing system, exploring policies and trust relationships followed by management of data and queries.

### A. Policies and Trust Relationships

*1) Organizational Sharing Policy:* To respect Qualitative Evaluation, organizations must separate mechanism from policy. Serrano et al. discuss four major challenges to cyber security sharing [14]. Lack of organizational policy can sometimes

prompt organizations to depend on technical solutions alone to manage data sharing, but we caution that technical solutions alone often are limited, unable to share enough, and prone to share too much (the corollary of Technical Limitations); we encourage organizations to consider *both* technical *and* policy aspects of sharing. In addition, organizations often lack policies for sharing data with other sites, prompting releases, when they do happen, to be done haphazardly by individuals. This results in unknown and uncontrolled risk exposure for the organization. Before an organization participates in cross-site data sharing, we encourage them to consider how to manage this risk and define approaches for how the data is governed, exposed, protected, and stored long-term. We argue that a balanced combination of technical and policy methods are necessary to meet the goals of sharing given one's risk tolerance.

*2) Minimal Requisite Fidelity:* Fisk et al. introduced *minimal requisite fidelity (MRF)* to the field of steganography, which they defined as the minimal degree of signal fidelity that was acceptable to end users and destructive to covert communications [15]. We extend this concept to the field of privacy, where the MRF of an information transaction would be the minimal degree of trust, the minimal amount of data exchange, and the minimal amount of disclosure between data owner and requester such that the exchange is acceptable to the requester but minimizes leakage of personal information or data. To respect Least Disclosure, systems should use the MRF of an information transaction as a gauge to the inherent privacy of the transaction. This suggests the need to have multiple levels of responses. A response to a query can be a yes or no, or escalated to data about a specific IP address or time window, or escalated further to a more complete traffic snapshot.

For instance, an analyst at a remote site may want to know whether the string `EmH0t=.q` was seen in TCP connections on port 927, within a certain time window. Where a traditional system might respond with a list of matching packets, MRF suggests that the system reply with a mere "yes, I saw that." This terse reply limits disclosure of potentially sensitive data that may be also in the matching packets.

*3) Least Privilege:* Cyber security sharing systems have multilateral trust relationships, which add complications over traditional central repositories for data access when managing information exchange. The approach of *least privilege* is based on the concept of Minimal Requisite Fidelity and requires that participating organizations assign to each entity the minimal privilege level as required to meet the objectives of data sharing. Multiple privilege levels can correspond to accesses and permissions granted to queries and responses on various data; for example, a high privilege level might allow visibility into more sensitive parts of data. An organization will have many different levels of privilege with various organizations based on the level of trust between each pair, but the organization must ensure that each relationship is granted the minimal amount privilege required to achieve the transaction, and no more.

Defining privilege levels should happen internally and externally to an organization. Within an organization, internal users are given a set of permissions for data (raw or otherwise) and program execution upon that data. Externally, organizational entities will define a relationship and policy agreement on sharing and use of data. Enforcement of privilege is done via the architecture by managing and limiting the set of queries and corresponding responses.

### B. Data Management

*1) Data Confinement:* To respect Least Disclosure, organizations must first consider the exposure of their data, where more exposure ultimately equates to a higher risk of unintentional disclosure. Within this realm, we propose the concept of *data confinement*: while owners may be willing to answer questions about their data, they may be less willing to make wholesale copies of their data and thereby lose control over its dissemination. In § III-C we discuss techniques to control the level of detail in answering questions about the data, limiting data disclosure.

An analyst or researcher fundamentally needs to ask questions about datasets. Historically, the mechanism for answering questions was to download datasets and run whatever procedures are needed against the data locally, on the researcher's laptop or local server array. However, Least Disclosure and risks of de-anonymization suggest instead that researchers should be able to ask questions about datasets instead of revealing data wholesale. Data Confinement allows researchers and analysts to ask questions that will be answered by the site holding the data, obviating the need to transfer entire datasets.

For instance, Alice may be curious if Bob's name servers have been getting queries for hosts in the `example.com` domain. Rather than sharing full logs (which contain information about other domains), Bob may instead support an API that allows Alice to ask "Are there any records that match `example.com`?" or "What are the records that match `example.com`?".

*2) Secure Data Archive:* Organizations must also consider how their data is stored. Unlike end-to-end encryption, where the source and destination of data is explicitly known, and full-disk encryption, which is generally transparent to applications, a secure data archive must manage the challenges of long-term storage with multiple potential users of data. It must thus consider encryption of data-at-rest, but also key-rollover and aging (to be robust over the long term), and access control and access auditing.

*3) Anonymization:* Anonymization is frequently used to sanitize data before release in such a way that any personal information or data is obfuscated or removed. This technique is very useful to researchers, but robustness to disclosure often comes with reduced utility of the remaining data. Moreover, attacks on anonymized data can be subtle [16], and released data must be robust to both current and *future* attacks. Additionally, in recent years several flaws have been identified by the community, and illustrated when researchers were able to re-identify "anonymized" users in the publicly released AOL search and Netflix Prize data. Ohm [13] provides an overview of anonymization of released datasets and its failures. Narayanan *et al.* [12] and Schneier [11] have also explored the deficiencies in current anonymization techniques. Coull *et al.* [17] describe techniques to infer network topology and de-anonymize servers in anonymized network traffic datasets. Due to these identified challenges, while storage must include careful and validated anonymization, by itself anonymization is insufficient. Instead, anonymization must be a component of a multi-pronged data protection approach.

*4) Data Aging:* Lastly, organizations must consider their risk tolerance in terms of long-term data storage. As recognized by Perlman [18], retention of data (or even "deleted" data) can be viewed as a new risk beyond simple media cost. *Data Aging* suggests that after a certain time window, data

should be reduced into a new format. This new format may take less storage space (addressing practical concerns) and removes sensitive information for risk management. Limited retention, a similar concept, is to make data available for a specific amount of time before permanently making it unrecoverable; data aging would be utilized when longitudinal studies over long-term data is needed.

Data Aging is often part of organizational e-mail policies; we suggest it should be part of a security archive. Kornexl et al. discussed one technique for Data Aging [19] for storage and lookup efficiency in network packet recording and retrieval. By removing sensitive data over time (perhaps when data moves from near-term storage to long-term archive), liability is reduced should problems occur.

For example, DNS queries, initially stored as complete raw packet captures, could be reprocessed to store daily summaries of lookups after some retention time set by policy and storage constraints (e.g., raw data older than 30 days is reprocessed). Rather than storing the IP and timestamps of every lookup indefinitely, an archival program could instead summarize daily counts for each host lookup, omitting who did the lookups and its timing. This reduces the redundancy of storing the same answer for each request (partially inspired by multi-level flow archives [20]) and protects users' privacy by removing sensitive information.

### C. Query Management

To realize Inquiry-Specific Release, a system must require query/response transactions on confined data stored at the source instead of providing whole datasets for arbitrary analysis. Within this concept we discuss the approach of requiring *moderated queries* on stored data. Using such queries, organizations can limit their trust of external organizations, restrict the types of queries that are allowed on the stored data, and *control the disclosure* of their data through query rate limiting. Additionally, organizations should ensure that they only query other data sources in ways that preserve their privacy and therefore reduce their exposure using *poker queries*.

Privacy Balance is illustrated with query management. As an example, an analyst interested in the retrieval of a specific URL could ask for that specific URL (maximum querier disclosure); she could ask for all URLs with a key substring, and then filter out only the ones of interest; or she could ask for every HTTP log in a time window (maximum responder disclosure).

*1) Moderated Queries:* Consider a cooperative cyber defense system in which two organizations are willing to disclose some information to each other in order to defend against mutual threats. Organization A performs anomaly detection on network event data and identifies a set of anomalous events of concern. For each of those events, A wishes to ask Organization B whether or not it has seen the same activity as a historical or new anomaly. Least Disclosure suggests a system will benefit from *moderating queries* by giving them structure that makes them easier to reason about. Structured queries would make use of an existing query language (e.g., SQL) and could be further restricted to a subset of clauses or operators to guarantee policy or privacy constraints. For example, a network traffic data system could allow only queries as tcpdump (BPF) expressions [21], in specific formats like Snort signatures [22], in specifically designed, constrained query languages [23], or with explicit models for privacy leakage that satisfy differential privacy [24], [25]. These limited query languages may allow users to ask specific questions without accessing complete data.

*2) Poker Queries:* When utilizing a system that relies on queries, requesting organizations must be concerned with what information is disclosed simply by querying another site regarding a security incident. If an organization issues queries on a specific security issue, they could inadvertently disclose that they had been compromised. In the American card game of poker, an important strategy is to not let fellow players guess what cards you hold while still taking public actions to improve the cards in your hand. We therefore define a *poker query* as a query where you minimize what you disclose when making your query. (A zero-knowledge protocol is a similar concept, but poker queries do not include a challenge-response.)

This problem has been acknowledged and studied in the community. Various tools and solutions that one could use to protect privacy in queries include (but are not limited to) salted hashes [26], homomorphic encryption [27], [28], privacy integrated queries [24], privacy preserving queries [29], location cloaking [30], distributed noise generation [31], and secure queries [23]. By designing a system that includes support for poker queries, analysts and researchers can ask questions of a dataset without revealing excessive information about why they want to know.

For example, researcher Alice may want to know if others have fallen prey to an intrusion with a specific network signature. She may also not be comfortable revealing that her organization has been infiltrated, nor that there is an investigation taking place. By using a poker query, Alice can discover if the attack specifically targeted her organization or if it is part of a larger, unfocused attack on multiple domains without revealing sensitive information about her organization.

*3) Controlled Disclosure:* To meet our goals of both Least Disclosure *and* Forward Progress, tools for cyber security sharing must rate-limit queries and responses by assigning privacy allotments to each organization [24] in order to mitigate the risk of privacy diffusion and secondary privacy damage [5], data query correlation [32], and other related exploits. Additionally, systems must restrict the kinds of questions that can be asked. In this sense, we can limit the specificity of questions that systems will answer: general questions are better than specific ones. For example, answering how many DNS lookups were made for `host.example.com` on a given date exposes less information than answering which IP addresses looked up `host.example.com` on that date.

## IV. Conclusion

This paper discussed the privacy challenges of systems that share cyber security information across multiple organizations. To guide the design of such systems we presented three privacy principles: Least Disclosure, Qualitative Evaluation, and Forward Progress. To make these principles more concrete, we then discussed how they apply to reduce risks of the data exposure and help manage trust requirements for data sharing. Our goal is to balance privacy, organizational risk, and the ability to improve response to security events. We are currently working to implement these approaches in our system for cyber security sharing, Retro-Future, to demonstrate how a system can both protect privacy and share data effectively.

## REFERENCES


[1] National Science Foundation, "Award and Administration Guide," Section VI.D.4.b. [Online]. Available: http://www.nsf.gov/pubs/policydocs/pappguide/nsf15001/aag_6.jsp

[2] United States. Cong. House of Rep. 114th Congress, "H.R.234 - Cyber Intelligence Sharing and Protection Act," 2015, Introduced in the U.S. House of Rep; 8 Jan 2015. [Online]. Available: https://www.congress.gov/bill/114th-congress/house-bill/234

[3] Open Knowledge. (2015) Open Definition. [Online]. Available: http://opendefinition.org/

[4] D. Dittrich and E. K. (editors), "The Menlo report: Ethical principles guiding information and communication technology research," United States Department of Homeland Security, Tech. Rep., Sep. 2011.

[5] B. Krishnamurthy and C. Wills, "Privacy diffusion on the web: a longitudinal perspective," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 541–550.

[6] M.-Y. Huang, R. J. Jasper, and T. M. Wicks, "A large scale distributed intrusion detection framework based on attack strategy analysis," *Computer Networks*, vol. 31, no. 23, pp. 2465–2475, 1999.

[7] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer networks*, vol. 31, no. 23, pp. 2435–2463, 1999.

[8] "The VERIS Framework," 2015. [Online]. Available: http://veriscommunity.net

[9] S. Barnum, "Standardizing Cyber Threat Intelligence Information with the Structured Threat Information eXpression (STIX)," 2013.

[10] The National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research, "The Belmont report: Ethical principles and guidelines for the protection of human subjects of research," Department of Health, Education, and Welfare, Tech. Rep., Apr. 1979.

[11] B. Schneier, "Why 'Anonymous' Data Sometimes Isn't," *Wired*, December 13, 2007.

[12] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Security and Privacy, 2008. SP 2008. IEEE Symposium on*. IEEE, 2008, pp. 111–125.

[13] P. Ohm, "Broken promises of privacy: Responding to the surprising failure of anonymization," *UCLA L. Rev.*, vol. 57, p. 1701, 2009.

[14] O. Serrano, L. Dandurand, and S. Brown, "On the design of a cyber security data sharing system," in *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*. ACM, 2014, pp. 61–69.

[15] G. Fisk, M. Fisk, C. Papadopoulos, and J. Neil, "Eliminating steganography in internet traffic with active wardens," in *Information Hiding*. Springer, 2003, pp. 18–35.

[16] R. Pang, M. Allman, V. Paxson, and J. Lee, "The devil and packet trace anonymization," *ACM SIGCOMM CCR*, vol. 36, no. 1, pp. 29–38, 2006.

[17] S. E. Coull, C. V. Wright, F. Monrose, M. P. Collins, M. K. Reiter *et al.*, "Playing devil's advocate: Inferring sensitive information from anonymized network traces." in *NDSS*, vol. 7, 2007, pp. 35–47. [Online]. Available: http://www.internetsociety.org/sites/default/files/coull_1.pdf

[18] R. Perlman and R. Perlman, "The ephemerizer: Making data disappear," *Journal of Information System Security*, vol. 1, pp. 51–68, 2005.

[19] S. Kornexl, V. Paxson, H. Dreger, A. Feldmann, and R. Sommer, "Building a time machine for efficient recording and retrieval of high-volume network traffic," in *ACM IMC 2005*. USENIX Association, 2005, pp. 23–23.

[20] L. Quan and J. Heidemann, "On the characteristics and reasons of long-lived Internet flows," in *ACM Internet Measurement Conference*. Melbourne, Australia: ACM, Nov. 2010, pp. 444–450.

[21] A. Begel, S. McCanne, and S. L. Graham, "BPF+: Exploiting global data-flow optimization in a generalized packet filter architecture," in *ACM SIGCOMM Conference*. Boston, Massachusetts, USA: ACM, Aug. 1999.

[22] M. Roesch *et al.*, "Snort: Lightweight intrusion detection for networks." in *LISA*, vol. 99, 1999, pp. 229–238.

[23] J. Mirkovic, "Privacy-safe network trace sharing via secure queries," in *Proceedings of the 1st ACM workshop on Network data anonymization*. ACM, 2008, pp. 3–10.

[24] F. D. McSherry, "Privacy integrated queries: an extensible platform for privacy-preserving data analysis," in *ACM SIGMOD 2009*. ACM, 2009, pp. 19–30.

[25] C. Dwork, "Differential privacy: A survey of results," in *Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[26] A. D. Kent and L. M. Liebrock, "Secure communication via shared knowledge and a salted hash in ad-hoc environments," in *Computer Software and Applications Conference Workshops (COMPSACW), 2011 IEEE 35th Annual*. IEEE, 2011, pp. 122–127.

[27] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009.

[28] E. Mykletun and G. Tsudik, "Aggregation queries in the database-as-a-service model," in *Data and Applications Security XX*. Springer, 2006, pp. 89–103.

[29] F. Emekci, D. Agrawal, A. E. Abbadi, and A. Gulbeden, "Privacy preserving query processing using third parties," in *ICDE'06*. IEEE, 2006, pp. 27–27.

[30] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*. ACM, 2003, pp. 31–42.

[31] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Advances in Cryptology-EUROCRYPT 2006*. Springer, 2006, pp. 486–503.

[32] V. Sharma, G. Bartlett, and J. Mirkovic, "Critter: Content-rich traffic trace repository," in *Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*. ACM, 2014, pp. 13–20.