SCRAM: A Platform for Securely Measuring Cyber Risk

Leo de Castro¹ Andrew W. Lo^{2,3,4,5,6} Taylor Reynolds^{2,7}
Fransisca Susan³ Vinod Vaikuntanathan^{2,5,8} Daniel Weitzner^{2,8,7}
Nicolas Zhang²

¹Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America,

²Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America,

³Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America,

⁴Laboratory for Financial Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America,

⁵Department of Electrical Engineering and Computer Science, Schwarzman College of Computing, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America,

⁶Sloan School of Management, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America,

⁷Internet Policy Research Initiative, Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America,

⁸MIT-IBM Watson AI Lab, International Business Machines and Massachusetts Institute of Technology, Cambridge, Massachusetts, United States of America

Published on: Sep 16, 2020

DOI: <https://doi.org/10.1162/99608f92.b4bb506a>

License: [Creative Commons Attribution 4.0 International License \(CC-BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

ABSTRACT

We develop a new cryptographic platform called SCRAM (Secure Cyber Risk Aggregation and Measurement) that allows multiple entities to compute aggregate cyber-risk measures without requiring any entity to disclose its own sensitive data on cyberattacks, penetrations, and losses. Using the SCRAM platform, we present results from two computations in a pilot study with six large private-sector companies: (1) benchmarks of the adoption rates of 171 critical security measures and (2) links between monetary losses from 49 security incidents and the specific sub-control failures implicated in each incident. These results provide insight into problematic cyber-risk-control areas that need additional scrutiny and/or investment, but in a completely anonymized and privacy-preserving way.

Keywords: cyber risk, cybersecurity, cyber losses, cybersecurity policy, cybersecurity metrics, multiparty computation, privacy preserving risk computation

Media Summary

Criminals benefit whenever their victims hesitate to share information about a crime or refuse to give details about its surrounding circumstances. There are legitimate reasons for this nondisclosure, but a lack of information-sharing allows criminals to use their past methods of success on new, unsuspecting victims. This is the current, and somewhat surprising, state of affairs for companies experiencing a cyberattack. Firms that share information about their attacks may reveal sensitive information to their competitors, open themselves up to litigation, and damage their own reputation in ways that could be even more costly than the initial attack. The long-term result of this standoff is that cyberattacks happen routinely but we learn very little about them collectively because firms are reluctant to share what happened.

In the past, the only way to aggregate and share information about cyberattacks was through a trusted third party. Every affected party would share private details of their attacks and the size of their losses with this trusted aggregator, who pledged to keep the data private and only release aggregated summaries and loss statistics. There are a number of problems with this approach. First, some of these trusted third parties have also been victims of cyberattacks that have put the data of all the submitting companies at risk. Second, many firms still refuse to share their most sensitive losses with a third party for fear of accidental disclosure, reducing the effectiveness of these intermediaries.

To address this longstanding cybersecurity challenge, we describe a new platform called SCRAM (Secure Cyber Risk Aggregation and Measurement) that uses new cryptographic tools to compute aggregate statistics without ever requiring a firm to disclose its own attack and loss data to anyone else—even to a third party. At an abstract level, this is made possible by the ability to perform mathematical computations on encrypted data that cannot be read or unlocked by the computing agent. Leveraging these mathematical properties, we can

gather locked data from firms, use our platform to calculate a locked result, and then ask each contributing firm to help unlock only the computed aggregate using their own secret cryptographic key. If any contributing company decides to back out and not unlock their portion of the answer, the result will stay securely locked beyond the reach of all parties involved. The power of this platform is that it allows firms to contribute locked data that would otherwise be too sensitive or risky to share with any third party.

To test this platform, we invited seven large companies (each with over \$1 billion in annual revenues) to contribute encrypted information on cyberattacks they have experienced to our platform. This served a practical purpose as well; their contributions include information about their network defenses, which is suitable for benchmarking, as well as a list of all monetary losses from cyberattacks and their associated defensive failures over a 2-year period. The results of our cryptographic aggregation process provide new insights into the adoption rate of defenses across this group of companies, and new information about the defensive failures that led to the largest monetary losses. These findings should help all firms direct their investments in cybersecurity to defenses that have the highest return.

1. Introduction

Our economy and society increasingly depend on our ability to defend data and information-technology infrastructure. Despite this dependence, we only have a rudimentary understanding of how to protect these systems and what defensive failures result in successful attacks. Measuring cybersecurity risk and the material impact of events has been a long-running challenge ([Cavusoglu et al., 2004](#)). Attacks are increasing over time ([U.S. Council of Economic Advisers, 2018](#)), and useful insights into the causes of successful attacks could help all firms improve their defenses. However, firms currently have little incentive to report attempted or successful attacks. In fact, sharing such sensitive information could invite regulatory scrutiny or create reputational harm for the company, and many are understandably concerned that making their data public could provide a competitive advantage to other firms ([Bandyopadhyay et al., 2009](#); [Swiss Re Institute, 2017](#)). The result is an environment where attacks happen on a regular basis, but collectively we learn very little from them. Ultimately, we need to know which defensive measures (that is, controls) actually work, and which control failures lead to the largest losses as a way to direct scarce security funds toward the defenses that will have the largest levels of risk reduction and return on investment. These insights will remain out of reach until firms are willing to contribute sensitive data about their losses and defensive strategies.

In this article, we present a solution to this challenge with the creation of a new cryptographic platform called SCRAM (Secure Cyber Risk Aggregation and Measurement). SCRAM can aggregate sensitive but encrypted data from multiple firms without requiring any firm to disclose their own data to others. The SCRAM platform represents an entirely new type of data collection and analytic tool for data scientists to use for measuring cybersecurity risks. This platform will improve security for all firms by allowing researchers to extract the insights from highly sensitive data that firms would otherwise not be willing to share.

SCRAM employs several known cryptographic techniques to ensure that data remain private throughout the computations on the platform and is never seen outside of the contributing party. We build upon ideas put forward by [Abbe et al. \(2012\)](#) and [\(Asharov et al., 2012\)](#).

1.1. Major Contributions

This work makes three major contributions to the literature. First, we provide a proof of concept using real data from large firms to show how a platform such as SCRAM can use multiparty computation (MPC) to securely calculate aggregated measures over sensitive cybersecurity data. We build upon previous research that has used MPC to run a double auction for the Danish sugar beet market ([Bogetoft et al., 2009](#)); securely link Estonian education and tax databases ([Bogdanov et al., 2016](#)); develop a proposed method for protecting privacy in large-scale genome-wide association studies ([Kamm et al., 2013](#)); run a simulation of a decentralized and privacy-preserving local electricity trading market ([Abidin et al., 2016](#)); and perform an analysis of the gender wage gap in Boston using data from a large set of Boston employers ([Lapets et al., 2016](#)).

Second, we make substantial progress in answering the three long-standing challenges for security practitioners laid out by Dan Geer ([Geer et al., 2003](#); [Hubbard & Seiersen, 2016](#)): understanding how secure a company is, knowing how their security posture compares to their peers, and evaluating whether they are spending the right amount of money on security controls. Our computational platform provides preliminary information to a subset of large firms with sophisticated cyber defenses about which security defense failure categories are leading to the largest firm losses, helping them evaluate and guide their security investments. We also provide previously unavailable benchmarking data so that firms can compare their security posture against their peers. While our sample size is small and includes only firms that were willing to participate, we are still able to extract some characteristics about the distribution of losses and identify the initial control failures that lead to high losses in these firms. Once identified, we can put additional focus on these problematic areas in future rounds. This preliminary work sets the stage for a much larger set of firms to examine how they defend themselves and determine which control failures have led to the largest losses, which will allow us to begin calculating a return on investment for various defensive measures.

Third, we illustrate some of the practical issues associated with pooling and computing with encrypted data. We share the specific challenges we encountered throughout the data preparation and computational process to the members of the broader data science community who may be considering encrypted computations such as these in the future.

1.2. Related Work: Secure Multiparty Computation and Its Applications

Secure MPC refers to the problem where n parties, each with their private input x_i , come together to compute an agreed-upon function f on the collection of their inputs, that is, $f(x_1, x_2, \dots, x_n)$. The problem was formulated by [Yao \(1986\)](#) in the two-party setting, and by [\(Goldreich et al., 1987\)](#) in the multiparty setting. These works also showed a remarkable result, namely, that any function whatsoever can be computed in this

manner while revealing to the parties only $f(x_1, \dots, x_n)$ and nothing else. Since then, there has been over three decades of fruitful research on more efficient and secure protocols.

These classical works have a few major drawbacks when applied to our setting. The first is that they require a large amount of network communication relative to the number of parties and the size of the computed function f . While, in principle, we could connect the participant machines by a high-throughput network, this is a logistical hurdle in practice. The second drawback is that the computational work needed to securely compute the function is uniformly distributed among the participants. This means that as the size of the function grows, the computational work required from *each* participant grows proportionally.

More recently, threshold homomorphic encryption has emerged as a particular way of constructing a secure MPC protocol ([Asharov et al., 2012](#)). This approach is attractive for our setting because it addresses both issues described above. In particular, the amount of network communication only needs to scale with the size of the inputs to the function, rather than the size of the entire function. In addition, the bulk of the computation only needs to be performed by a single machine. This asymmetrical workload is very desirable in our setting since, even when securely computing large functions, it allows the participants to have weak machines that send the input to a powerful, central machine and then later receive the result from this central machine. We refer the reader to Section 2.3 for a more in-depth discussion of this MPC solution.

In the last decade, many MPC tools and solutions have been optimized and implemented by several libraries. These include the secure MPC libraries of the Boston University Accessible and Scalable Secure Multi-Party Computation project ([Boston University, n.d.](#)), the homomorphic encryption libraries of Microsoft SEAL ([Microsoft, 2020](#)), and Duality’s PALISADE ([Polyakov et al., n.d.](#)).

Finally, these tools have been applied to solve problems in several domains. We mentioned some of the most prominent results that build real systems and solve a compelling real-world problem in Section 1.1.

More recent related work in the area of applied secure aggregation includes the two-party computation protocol of [Ion et al. \(2019\)](#), which securely performs a set intersection protocol between two parties, and the privacy-preserving browser data aggregation system of [Corrigan-Gibbs & Boneh \(2017\)](#).

1.3. Target Audience

The primary target audience of this research is the group of security decision makers within a firm who are charged with deploying scarce security resources to protect information assets, for example, the chief information security officers (CISOs), as well as the policymakers and regulators with governmental responsibility for cybersecurity. The research will also be of interest to the members of the broader data science community who are grappling with ways to compute with sensitive data sets without putting the underlying data at risk of disclosure. Our approach is generalizable over a broad set of applications.

1.4. Road Map

In Section 2, we describe the SCRAM platform and introduce the cryptographic tools that underpin the secure computation process. We also explain the process of bringing in participants, the development of the data input structures, and the preparation of the data for secure computation. Section 3 provides the results of two computations on 49 security incidents contributed by six large firms. Finally, Section 4 summarizes the implications of this research for the security community and data scientists. We conclude with a discussion of potential next steps for further research.

2. Study Designs

In this section, we define our objective and then explain the design, implementation, testing, and operation of our secure computation platform.

2.1. Definition of the Problem

Currently, the best aggregated data available on cyber risk and its associated losses come from anonymized data pools ([Operational Riskdata exchange association \(ORX\), 2017](#); [SAS Corporation, 2015](#)) and research surveys ([Advisen, 2019](#); [Anderson et al., 2019](#); [Anderson et al., 2013](#); [Eling & Wirfs, 2019](#); [NetDiligence, 2018](#); [Ponemon Institute, 2016](#); [Powers, 2007](#); [Verizon Corporation, 2017](#)). These existing pools provide valuable, if imprecise, data on market risk, operational losses, and insurance performance. However, existing pools and surveys only include information that firms have deemed safe to disclose to an outside party. They lack the most sensitive data that would most significantly improve our understanding of cyber risk. This research is an attempt to overcome this challenge.

In 2015 and 2016, MIT's Internet Policy Research Initiative (IPRI) held a series of sector-specific workshops focused on protecting critical infrastructure. The workshops included presentations by CISOs from four distinct economic sectors (electricity, oil and gas, finance, and communications) who discussed the challenges they faced securing and defending their networks ([Brenner, 2017](#)). A common theme began to emerge across all four sector-specific meetings. The CISOs stated that deploying security controls was akin to “investing in the dark,” because they lacked the necessary illumination into the defensive postures and related losses of other firms that would only be available if firms shared information. Despite this, the CISOs of these firms were also reluctant to share information because of the sensitive nature of their own data.

To address this challenge, we brought together a multidisciplinary team of specialists in financial risk management, cryptography, and computer security from across MIT to design and build a new platform using cutting-edge cryptographic techniques that could be used to securely and privately calculate aggregated metrics on cyber defenses and loss data, without requiring firms to disclose their own data. This new SCRAM platform would provide clarity and visibility on how firms as a whole defend themselves, and improve the understanding of the relationship between control failures and financial losses.

2.2. General Approach

The SCRAM platform supports secure and private aggregation pools for data, building upon ideas put forward by [Abbe et al. \(2012\)](#) and [\(Asharov et al., 2012\)](#). We assume a data set is distributed across the participants, with each participant contributing a subset of the data entries into the pool. Using MPC, we compute fixed functions of the input data without revealing the individual entries to anyone other than the participant who contributed them. MPC is a general-purpose tool that allows participants who submit private inputs to a known mathematical function to learn the evaluation of the function on their private inputs without revealing their private inputs to the other participants. Our particular implementation of MPC is optimized for large, tabular data sets. Technical details of our SCRAM platform, including these optimizations, are described in the Appendix, and the source code is available at <https://github.com/CSAIL/ipri-scram>.

These MPC techniques allow us to maintain the accuracy of our computations without sacrificing the privacy of the participants, a major benefit over other approaches such as differential privacy. The main drawback of MPC is that it is completely agnostic with regard to the function. It will faithfully produce the output of whatever function it is programmed to compute. MPC provides no guarantee that the function output will not itself reveal information about private inputs. This risk is mitigated by the fact that the function must be agreed upon by all participants in order for the output to be learned by any participant, but there are still ways in which a seemingly benign function can reveal information about a participant's input. One example considered in this project is maintaining the privacy of a firm with a large outlier incident when computing the average of the input values. This is discussed in Section 2.8.

2.3. An Introduction to MPC and the SCRAM Platform

Traditionally, when members of a group want to learn about trends across their membership without revealing information to other members in the group, they need to select a trusted third party. This third party will gather data from all the participants, pool it, run computations on the data to produce summary statistics and analysis, and then send only the results back to the participants in the pool. The third party needs to be trusted because it can view the data sent in by all the participants: Individual participants cannot see the inputs of other contributors, but the trusted third party can see everything. The process works well when participants are comfortable sharing their information with the third party (a risk) in exchange for learning more about the dynamics of the group (a benefit).

But what if participants are unwilling to reveal sensitive data to even a trusted third party? MPC offers the same functionality as the data pool described above, but without requiring a trusted third party to see the data. This is possible due to a combination of the mathematical properties of encrypted data and clever structuring of the computations.

SCRAM mimics the traditional aggregation technique, but works exclusively on encrypted data that it cannot see. The system takes in encrypted data from the participants, runs a blind computation on it, and returns an

encrypted result that must be unlocked by each participant separately before anyone can see the answer. The security of the system comes from the requirement that the keys from all the participants are needed in order to unlock any of the data. Participants guarantee their own security by agreeing to unlock only the result using their privately held key.

Cryptographic tools such as MPC and public key cryptography provide a way to perform mathematical operations on encrypted data without ever exposing the underlying data. While there are a variety of solutions to the challenge of secure computation, we choose an approach for SCRAM that provides simple, straightforward security guarantees as well as support for complex computation. Each of SCRAM's steps are provided in Figure 1, and further technical details are given in the Appendix.

The SCRAM computation platform consists of three main elements: a central server, software clients, and a communication network to pass encrypted data between the clients and the server. The central server manages the data collection from the clients and hosts the core cryptographic software that performs only predefined and approved computations on the encrypted data. It is also responsible for piecing together and redistributing the joint public key that is used by the clients to encrypt any data for the computation.

- (1) Each firm individually generates its own key pair, where each key pair contains a public encryption key and a private decryption key.
- (2) All firms submit their public keys to the server.
- (3) The server combines all firms' public keys into a single joint/shared public key.
- (4) This new joint/shared public key is distributed from the server to all firms.
- (5) Each firm encrypts its private data using this new joint/shared public key, generating a ciphertext (an encrypted block of data).
- (6) Each firm sends the ciphertext of its private data to the server. This ciphertext completely hides the firm's data.
- (7) The server runs computations on all the encrypted data, producing an encrypted result of the computation.
- (8) The server sends the encrypted result back to each firm.
- (9) Each firm uses the private key they generated in Step 1 to partially decrypt the answer.
- (10) Each firm sends this partially decrypted answer back to the server. Note that without all the partial decryption pieces from all firms, the result is still completely hidden.
- (11) The server combines the results of all the partial decryptions it receives from firms to produce the decrypted result that is then shared with all firms.

Figure 1. Principal steps of the MPC implementation in SCRAM.

The individual software clients create public/private key pairs on the local machine and enable the input of data, its encryption, and the communication back and forth with the server. The individual clients also play a vital role in decrypting the encrypted result of the computation run by the server. If firms wish to have an additional security guarantee, they can compile the joint public key for the computation individually by using the public keys of every other participant as well as their own. This ensures that the joint public key was generated correctly, that is, the data encrypted with this key can only be decrypted if the firm's own private key (which it controls) is used in the distributed decryption process.

The computation requires a network to pass information back and forth to perform the operations on the encrypted data. The only data ever transmitted over the network are either strongly encrypted or the public key of the clients, which can only be used to encrypt data, not decrypt it. The network uses TLS (Transport Layer Security) for a second layer of encryption to protect the encrypted data in transit.

Once we completed constructing the platform, we ran a series of internal tests with synthetic data to ensure that the results emerging from its computations were consistent with the known inputs. These tests included a variety of data edge cases (e.g., large data files, data with significant outliers, large numbers, inputs with decimals, and incorrectly coded data) in order to test the consistency of the calculations and understand possible failure modes. The platform successfully passed all of the tests.

There are many approaches to implementing MPC that differ significantly from our chosen method. In this section, we briefly discuss some of the trade-offs we considered when selecting our MPC approach. This section can be safely skipped for readers who are not interested in these details.

When designing an MPC, one must frame it as a circuit that will be executed on the encrypted inputs to produce an encrypted output. The first decision that must be made is the type of circuit the MPC must support. In general, there are two principal choices: arithmetic circuits and Boolean circuits. Arithmetic circuits consist of gates that perform arithmetic operations, usually addition and multiplication. Boolean circuits, on the other hand, consist of Boolean gates, which operate on binary values. Common Boolean gates are AND, OR, and NAND gates. It is important to note that both representations are equivalent in the types of circuits that can be expressed; however, there are differences in the efficiency of these representations depending on the desired functionality.

Since arithmetic circuits allow arithmetic operations to be done in a single gate, they are a natural choice for calculations that require many arithmetic operations. The downside to arithmetic circuits is that several desirable operations, such as division and comparisons, are more expensive than their Boolean counterparts. Boolean circuits, in contrast, have a much more expensive method of implementing addition and multiplication, but can perform operations such as division and comparisons with greater relative efficiency. For the SCRAM platform, we chose to implement an MPC technique that supported arithmetic gates since a large majority of the operations in our computations were arithmetic. In the future, we intend to explore hybrid

solutions that allow for the benefits of both types of circuit representations, but the development of efficient hybrid solutions remains an active area of research.

Another important way that MPC techniques differ from one another is in the distribution of the work to perform the computation among the participating parties. In classical MPC constructions ([Goldreich et al., 1987, 1991](#)), the work of the MPC protocol is evenly distributed between all participating parties. The classical case is not ideal in our scenario, since we are working with participants in secure environments where it may be challenging to install new software on their internal machines. The solution we chose was to distribute inexpensive machines with our software already installed to each participant, and to select an MPC method that minimized the work required by these commodity machines. To compensate for this large number of weak participants, we chose a central server with enough computational power to run all of the MPC protocol. The distribution of work in the protocol is very uneven; almost all of the work is happening on one machine. In addition, the work done on the participants' machines is largely independent of the actual computation done on the encrypted data, which allows us to more easily update and maintain the SCRAM software.

The MPC approach in our implementation is based on threshold homomorphic encryption, and the particular homomorphic encryption scheme that we use supports arithmetic circuits. Solutions based on homomorphic encryption have the advantage of supporting the asymmetrical workload discussed above. More details are given in the Appendix.

We conclude this section with a brief note on security. An astute reader will have noted that we trust the server to execute the correct computation in order for the full protocol to be correct. It is important to note that while a compromised server may calculate the wrong function, it cannot compromise the security of the encrypted inputs. In addition, all of the computations we consider are deterministic, so the correctness of the server's output can be checked by any participant who also has the other participants' inputs. While we did not implement this more secure protocol in our current version, we plan to include this functionality in later versions for participants who are willing to commit to a more powerful machine (i.e., one that can run the computation itself) to ensure the correctness of the results.

2.4. Participants

We developed our first aggregate computations with the participation of large firms (over \$1 billion in annual revenues) that had a high level of security sophistication and a CISO. Firms of this size have the technological expertise and resources to work with us to design the appropriate questions and perform the internal data collection. The decision to focus on large and sophisticated firms also meant that outcomes from the computation would be more relevant to organizations with sufficient resources and need to make use of our results.

One key question we faced up front was whether to focus on a specific industry, or to run the aggregate computation on a broader cross-section of firms. The overwhelming preference of the CISOs was to select

firms across various sectors since security approaches in large companies are relatively similar, and many CISOs have worked in multiple sectors.

We recruited seven firms to work with us, across the health care, communications, retail service, and financial sectors. The participating firms had an average annual revenue of \$24 billion (median of \$18 billion) and an average of 50,000 employees (average and median). We invited a larger number of firms to participate, but some companies were cautious about putting sensitive data into a new platform and preferred to wait to see the initial results before committing to participate in future rounds.

This highlights an important issue with selection bias that may affect the general applicability of these findings. The firms that were willing to participate in the first practical use of the SCRAM platform have highly sophisticated security teams who understand MPC well enough to trust the computation and submit their data. One team, in fact, sent a cryptographer on staff to monitor the computation. Our results here should be viewed primarily as a proof-of-concept showing the potential of the SCRAM platform, with any conclusion about security from our results viewed through the lens of this subset of firms.

2.5. Developing the Questionnaires

We decided on two aggregate computations for the first use of the SCRAM platform: one to benchmark firm defenses against cyberattacks, and another to associate monetary losses to control failures. We worked closely with the participating companies over 4 months to develop the two computations and their associated input formats. The data input structures are explained below.

1. **Benchmarking:** In the first computation, we benchmarked the adoption of a broad set of cyber defenses to allow firms to compare their own security posture to the adoption levels across the group. We used the Center for Internet Security's (CIS) list of 171 critical security sub-controls that are meant to help organizations better defend against known attacks by distilling security concepts into actionable controls. The benchmarking consisted of a questionnaire in which firms indicated if they currently implement each of the 171 CIS sub-controls (Yes = 1, No = 0). We summed the results and divided by the number of firms participating in the computation to obtain the adoption rate of each sub-control across all participants. We considered other frameworks such as NIST 800-53 and MITRE ATT&CK for this benchmark, but we ultimately selected CIS because of the manageable size of the control list and the quantitative and binary nature of the responses, which made them easier to standardize across firms. We specifically chose to start with control frameworks at a granular level instead of program frameworks (ISO 27001, NIST CSF) or risk frameworks (NIST 800-39, FAIR), which have broader results but are more difficult to aggregate across firms. However, it should be stated that our SCRAM platform is capable of running computations using nonbinary inputs or other security frameworks.
2. **Linking monetary losses to failed sub-controls:** In the second computation, we gathered data on losses and their implicated security control failures in order to identify problematic security controls across the

group. We asked firms to submit lists of individual losses and indicate which sub-control failures were responsible. For the computation, the firms submitted a table with individual incidents on each row. Each participating firm assigned a monetary loss (in thousands, U.S. dollars) to each of their security incidents and then indicated up to five sub-controls (Yes = 1) that were responsible for each loss (either because it was in place and failed, or because it was not implemented). In this round, each implicated sub-control was assigned an equal proportion of the total loss during the computation. We implemented a minimum loss threshold of \$5,000 in order to exclude routine costs such as reformatting infected machines, and to focus specifically on larger incidents. The output of this computation was the total loss attributed to each sub-control across all submitted incidents.

2.6. Data Encoding

Anecdotally, data scientists spend up to 80% of their time finding, cleaning, and organizing data ([Bowne-Anderson, 2018](#)). However, MPC requires that all data cleaning is done by the contributors (privately) before data are encrypted and submitted for computation. With MPC, data analysts lose the ability to examine and clean the input data since it remains hidden even after the computation has finished.

This means that the SCRAM platform requires a significant time investment with data providers to standardize inputs and verify expectations in order to ensure contributors are submitting cleanly formatted data free of error. A client-side script was included in SCRAM to verify that data were formatted according to the agreed-upon standard, but these checks are only able to catch major formatting issues rather than more subtle errors. To compensate, we ran various encoding exercises to train participants on the process, uncover errors, and to harmonize inputs for the actual computation.

For example, we ran an exercise to examine the variance of encodings across firms by asking the participating firms to encode a well-known ransomware attack (Petya) and a generic distributed denial of service (DDoS) incident with the CIS sub-controls they believed would likely be implicated (up to five controls per attack). The results of this encoding exercise revealed that a much broader base of potentially problematic controls would likely be implicated than we had initially assumed. The five participating firms submitted a total of 19 unique sub-controls with responsibility in the hypothetical ransomware attack and 13 unique sub-controls in the DDoS attack.

- **Ransomware:** 19 unique sub-controls out of a total of 25 implicated (5 firms)
- **DDoS:** 13 unique sub-controls out of a total of 21 implicated (5 firms)

In contrast, if all firms had implicated the same sub-controls, only five unique controls would be implicated out of a total of 171. From this exercise, we learned that firms will likely view and encode the same incidents in different ways when they assign responsibility for incidents. This highlights the need to standardize the way firms record and encode incidents, which is a major direction for future work, as described in Section 4.4.

We also took steps to ensure that participants would fill out the distributed templates correctly. We asked participants to create a set of synthetic security incidents and encode them into our agreed-upon template, and then send it to us to evaluate. We found different interpretations of certain data fields and mistakes in the encoding of units, and reported our findings, with clarifications, back to all the participants before the date of the actual computation.

2.7. Summary Statistics Trade-offs

In any MPC-supported calculation, the choice of which summary statistics to reveal must be made with an eye toward the risk of data leakage from a single participant. Disclosing summary statistics can ‘leak’ certain information about the underlying data in cases where there are too few related observations. Since all results are revealed to all participants, the design of an MPC must account for the worst-case leakage that the MPC could reveal about the inputs. This is in stark contrast to nonencrypted computations, where researchers can choose *ex post* which results are safe to release in order to protect against disclosure of the underlying data.

Consider a concrete example. Suppose a group of participants wish to run an MPC where the input is a list of incidents, and each incident has a type and a cost. The output of this computation is a count of the number of incidents of each type and the total cost of each incident type. If there are many entries for each type of incident, then we can expect that the output of this computation will hide the input of any individual party. However, if there are only a small number of incidents of a certain type, then the output can potentially leak information about the inputs. The simplest case of this is if there is only one incident of a certain type. In this case, the output of the computation completely reveals the input entry corresponding to this incident type. The computation would reveal the exact losses associated with the single event, but not the identity of the submitter.

Now suppose the output of the computation included more sophisticated statistics about the cost of each incident type. If we include the standard deviation (second moment) of the costs of each incident type, then any incident type with up to two entries would completely reveal the input entries. This can be seen by writing out the equations for the count, sum, and standard deviation and then solving for the unknowns where there are only two input values. This analysis can be extended to higher moments of the input data; for example, if we also release the skewness (third moment) of the data, then any incident type with up to three entries completely reveals the original inputs.

One solution is to selectively withhold results that do not meet specified privacy thresholds. However, this requires computing the privacy requirements inside the MPC function, significantly increasing the complexity of the computation.

We approached this trade-off by evaluating which summary statistics are of the most interest to our objective and analyzing the cases in which input data are revealed. The following are our statistics of interest in descending order of importance:

- Sum of losses per sub-control
- Count of incidents implicating each sub-control
- Moment 1: Average loss per implication of a sub-control
- Moment 2: Variance of the losses for an individual sub-control
- Moment 3: Skewness of the losses for an individual sub-control

Table 1 enumerates the possible summary statistics we considered computing over the input data. Each summary statistic is based on observations related to a specific sub-control. We consider the *cumulative* leakage from revealing each summary statistic based on the number of observations that are summarized. For example, revealing the sum of the losses *completely reveals* all values if there are zero values (since the sum will be zero). If there is even one nonzero value in the sum, the inputs will be hidden. However, if the count is also revealed along with the sum, then it is necessary to have at least two observations to hide the input values because if there is only one observation, the count will identify this case and the sum will reveal the value. For this computation, we took a conservative approach and limited our reporting to sums, counts, and the average in order to prevent additional leakage.

Table 1. Leakage of summary statistics by number of observations.

Summary Statistic	Obs = 0	Obs = 1	Obs = 2	Obs = 3
Sum of losses	Revealed	<i>Hidden</i>	<i>Hidden</i>	<i>Hidden</i>
Count of losses	Revealed	Revealed	<i>Hidden</i>	<i>Hidden</i>
Average of losses	Revealed	Revealed	<i>Hidden</i>	<i>Hidden</i>
Variance of losses	Revealed	Revealed	Revealed	<i>Hidden</i>
Skewness of losses	Revealed	Revealed	Revealed	Revealed

There are several mitigating factors that help protect the privacy of the contributed data in the examples above. First, by using incidents rather than firms as the observational unit, we remove any linkages to a particular firm. For example, while the outputs could reveal a \$100,000 loss to a sub-control with a count of one, there are no data indicating the source company of the loss, or even how many other controls were implicated in the same incident. It is important to note, however, that this may not hold in the case of extreme outliers (see Section 2.8). Second, the summary output is only revealed within the group of participants, potentially lessening any impact of its disclosure.

It is worth noting that this leakage analysis does not consider additional information (e.g., a prior distribution over the inputs) that participants or parties who receive the computation output may have that could assist them

in recovering the inputs. This concern is particularly relevant in the case where a large outlier is present in the data set, especially if this outlier was publicly reported. Summary statistics that reveal disproportionately large losses can reveal which control failures and losses are attributed to the single outlier incident.

2.8. Outliers

MPC protocols are secure, but they do require careful application to avoid leaking sensitive information in results that could reveal information about the submitted data. One challenge emerging from our computations is how to deal with large outliers in the data, particularly because cybersecurity events are characterized by many small incidents, with a relatively small number of very large ones. An outlying security event that is much larger than any others can leak information about which sub-control failures led to that particular loss due to their disproportionately large magnitudes.

A simple example can illustrate how a single outlier can leak data. If a single loss is 100 times larger than any of the other 19 losses in a combined data set, then the proportion of the oversized loss attributed to each of the failed sub-controls in that particular event (up to five) will be larger than the sum of all other losses in the data set. This will effectively reveal the sub-control failures associated with the largest loss by their magnitudes.

One potential participant in our computation had an incident that may have been much larger than any of the other incidents submitted by other firms that we were aware of at the time. Including this outlier had the potential to leak information about which failures led to the losses of that specific large incident. Ultimately, we excluded the firm from our computations in this round, but it illustrates the need to develop a better solution so the firm can be included in future calculations.

There are two broad ways to address this potential leakage. The first and preferred method is to ensure that the computation data set includes a sufficient number of high-loss events (at least three) to mask which sub-control failures belong to which losses. These extremely high-loss events (e.g., significant breaches) are generally known to the public, so we should ensure that a sufficient number of firms with these large events are included in future rounds to mask any individual outlier's inputs.

The second method is censoring any outlier events that could leak data at the time of the computation due to a lack of comparable events. However, this removes some of the most valuable information from the data set, and limits our understanding of what failures lead to extreme losses.

We can include checks in future rounds that run precomputations on the data for significant outliers, and only allow the computation to run if there is sufficient diversity to cloak any extreme inputs. Ultimately, the best solution is increasing the sample size to ensure sufficient coverage of high-loss events. This is an important issue for research going forward.

3. Computation Results

This section presents the results of the two MPCs run with six firms on the SCRAM platform in April 2019, one on benchmarking firm defenses across the group and the other on mapping losses to specific control failures.

These results should be viewed as preliminary indicators of problematic controls rather than precise estimates of predicted losses that might be applied beyond the group. In the future, we hope to provide estimates with larger sample sizes that firms can use as part of their security investment decisions.

Each participant took around an hour to manually input data from their company machines into our dedicated, air-gapped terminals. Once the data were in our system, the computation took only seconds to run and send its results back to each of the firms present for the computation.

3.1. Computation 1: Defenses and Security Implementations

The results of the first computation are based on the questionnaire described in Section 2.5 and shown in Figures 2 and 3. Data from the 171 sub-controls were first grouped by the 20 broad CIS control categories in Figure 2. The data showed a relatively high rate of adoption across the different control groups, which should be expected from firms with this high level of security sophistication.

To put the level of sophistication of the participating firms in perspective, we analyzed adoption rates for sub-controls in our sample with CIS's own implementation groups, which divide the sub-controls into adoption categories for firms at different levels of security sophistication. The firms in our group adopted 91% of sub-controls in CIS Implementation Group 1, targeted at organizations with limited resources, 79% in Group 2 for organizations with moderate resources, and 63% of controls in Group 3 for organizations with mature resources.

In Figure 2, we can see that the adoption rates for control groups 13 and 14 are markedly lower than the others. Both averages are made lower by a low adoption rate of one sub-control in each group. Interestingly, one of these low-adoption sub-controls has monetary losses associated with it. Sub-control 14.05¹ (Utilize an Active Discovery Tool to Identify Sensitive Data) was only adopted by one firm, yet it was implicated twice as a control failure, resulting in \$187,000 of losses across the group.

Figure 3 displays the number of sub-controls at each level of adoption rate. The figure shows high adoption levels for most controls, but 12 sub-controls still have less than 50% adoption. One sub-control was not adopted by any firm. It is important to note that running this computation among a group of smaller or less sophisticated firms would likely result in lower adoption rates and a different distribution.

Figure 2. Adoption rate: Average sub-control adoption rates by CIS control group.

Figure 3. Adoption rate: Count of sub-controls by adoption rate among the firms.

3.2. Computation 2: Losses and Implicated Controls

The second computation links actual dollar losses of information security incidents to the sub-control failures that were deemed responsible for the incident by the submitting firm. We did not know how many incidents firms would contribute on the day of the computation. In the end, the six participating firms submitted a total of 49 incidents from 2017 and 2018 into the computation for a total loss of \$29.43 million, corresponding to an average of eight incidents per firm over the 2-year period and representing roughly 0.01% of the firms' total revenues over that time. The distribution of losses had many small incidents, intermixed with a few large losses. Log management issues, communications over unauthorized ports, problems with asset inventories, and a lack of well-functioning anti-malware software were the most problematic sub-controls. We provide more detail on each of these findings below.

Before the computation, we created a set of five broad loss 'buckets' at different loss thresholds to understand the distribution, while still keeping details of individual incidents private (see Table 2). We knew of one potentially large outlier because of a public disclosure, but ended up dropping that firm from the combined computation due to a formatting problem with its input data on the day of the computation.²

Table 2. Loss thresholds, U.S. dollars.

Category	Loss Thresholds
Low Loss	\$5,000 to \$50,000
Low/Mid Loss	\$50,001 to \$500,000
Mid Loss	\$500,001 to \$5,000,000
Mid/High Loss	\$5,000,001 to \$50,000,000
High Loss	\$50,000,001 and above

Figure 3 shows the number of incidents at the different loss thresholds. There were no losses in the two highest segments, while most incidents fell into the lowest loss category. These figures are descriptive of the firms who were part of our sample, but they could change significantly with different participants. A distribution can be seen from the results of many losses at the low end, and relatively few at the higher end of value.

These losses appear to follow a distribution suggested by (Eling & Wirfs, 2019), with a large number of small 'daily life' exposures along with a few 'extreme cyber-risk' incidents. Hofmann et al. (2019) and Wheatley et al. (2016) find similar results of heavy-tailed risk. It is important to note that none of the firms in our final computation had a large incident with tens or hundreds of millions of dollars of losses during the period. Even so, we notice a long-tailed distribution in the group, even in the absence of any extreme outlier.

Figure 4. Count of cybersecurity loss incidents in each loss threshold, by loss size.

When we ran the full computation, the security teams from the participating firms identified a total of 76 unique sub-controls that were responsible for all of the reported losses. These sub-controls represent just under half (44%) of all CIS sub-controls. Figure 5 shows a histogram of sub-controls by the number of times they were implicated in an attack. There are 39 sub-controls that were only implicated once. On the other hand, some sub-controls were implicated in up to 10 different incidents. Sub-control 5.01 (Establish Secure Configurations) was implicated 10 times, while 13.03 (Monitor and Block Unauthorized Network Traffic) and 18.10 (Deploy Web Application Firewalls) were both implicated 7 times (Figure 5).

Figure 5. Count of sub-controls by number of times implicated in a loss incident.

The computation identified a larger list of implicated sub-controls than originally predicted. This could be the result of a broad set of attack vectors or, more likely, differences in how firms view and assign responsibility to individual sub-controls. Sub-controls such as 5.01 that had 10 separate implications clearly represent a security challenge across firms, and these sub-controls deserve a deeper exploration of how their perceived failure leads to losses. Other sub-controls were implicated only once, and they may, in fact, be capturing the same phenomenon as failures in related sub-controls. Future work should examine ways to regroup sub-controls that may be implicated interchangeably.

Incidents are distributed unevenly across CIS control categories (Figure 6). The controls focusing on awareness and training (CIS 17), boundary defenses (CIS 12), and data protection (CIS 13) were the most commonly implicated in security incidents with financial losses. It is of interest to note that sub-controls related to security awareness and training (CIS 17) had the highest number of identifications. This seems logical enough, given the prevalence of phishing as an attack vector. This emphasizes the importance of training users to reduce the number of successful attacks. All of the top categories, however, deserve additional analysis to understand why they are so commonly implicated. Even if the financial losses are not large, the fact that they are involved in so many incidents within the group of firms highlights the level of risk they pose to organizations.

Figure 6. Incident appearance count, by CIS control category.

The computation results show significant differences in total monetary losses attributed to controls across different CIS categories. Figure 7 shows that audit logs (CIS 6), boundary defenses (CIS 12), hardware inventory (CIS 1), and malware defenses (CIS 8) had the largest total losses across the group. Audit logs (CIS 6) and application software security (CIS 18) ranked high on both number of incidents and total losses, and require a deeper examination in future rounds. The order of the losses by category are likely to shift with the introduction of additional firms to the computation, but the data are still useful to identify categories with high losses.

Figure 7. Sum of losses across 49 Incidents, by CIS control category, in U.S. dollars '000

The submitted data set allows us to look even deeper to the sub-control level to examine loss trends in our small sample. Seven sub-controls had attributed losses in excess of \$1 million (Table 3). The largest losses are linked to failures in log management, communication over unauthorized ports, and a failure of (or lack of) software to prevent malware attacks. The rankings are less important with our small sample size and may be the result of a smaller outlier, but the grouping of sub-controls with large associated losses is a valuable take-away for security professionals. A larger sample size of incidents will help us calculate the full scale of these problematic areas.

There are concrete lessons we can draw from our existing observations. First, network logs were implicated with the highest total losses of any sub-control. It is certainly a challenge to extract the right actionable insights out of logs, but logs are a detection measure rather than a protective one. Future work should examine why logs are attracting so much blame, at least in our sample. Second, the sub-control for denying communications over unauthorized ports (CIS 12.04) had the second-highest losses (\$4.5 million), yet 100% of firms in the benchmarking exercise implemented this sub-control. If this is the case, why is this sub-control still responsible for such high losses within our sample group? To security practitioners, port blocking may seem like a standard security measure with known and implementable solutions, yet it still accounts for high losses within our group.

Table 3. Largest losses by CIS sub-control.

CIS Num	CIS Name	Loss
6.05	Central Log Management	\$5.825 M

12.04	Deny Communications Over Unauthorized Ports	\$4.503 M
1.04	Maintain Detailed Asset Inventory	\$4.300 M
8.01	Utilize Centrally Managed Anti-malware Software	\$4.015 M
16.13	Alert on Account Login Behavior Deviation	\$1.632 M
6.06	Deploy SIEM or Log Analytic tool	\$1.575 M
20.02	Conduct Regular External and Internal Penetration Tests	\$1.350 M

The losses highlighted in Table 3 represent the largest losses in our sample, but there is a long tail of losses associated with the remaining 70 controls that can be seen in Figure 8 (subset with losses greater than \$100,000) and Figure 9 (full distribution). Again, a few controls have high losses, followed by a long tail of smaller ones.

Figure 8. Distribution of losses across sub-controls with losses of \$100,000 or greater, in U.S. dollars '000.

Figure 9. Distribution of losses across all sub-controls, in U.S. dollars '000.

4. Insights and Next Steps

Following the computation, we discussed the results with the participants for about 90 minutes. Several weeks later, we also held one-on-one conversations with each of the participants by phone to obtain additional feedback about the experience. Participants stated that they were pleased with the results. For many of the problematic sub-controls, the data served to confirm their own institutional intuition of problematic controls held across their peer group. In other cases, the firms were surprised by some of the findings, such as the high losses associated with log failures across multiple events. The data highlighted a potentially deeper need to extract better information from the company's IT logs. Participants said that the benchmarking exercise allowed them to compare their own internal protections with the averages across the group, but it was the monetary values on specific control failures that they found the most useful in terms of focusing their future security investments.

The three sections below describe in more detail several key insights drawn from the computational exercise and the data. Section 4.1 focuses on insights for the security community and Section 4.2 discusses implications for policymakers, regulators, and insurers, while Section 4.3 provides general lessons learned for the broader data science community. Finally, Section 4.4 highlights potential avenues for future work.

4.1. Implications for the Security Community

Several insights emerge from this research specifically for the security community. First, MPC can be used to pool very sensitive data in a way that enlightens the participants without sacrificing the security and privacy of

individual firm data. This is the first step toward a better understanding of the relationship between defensive measures and returns on security investment.

Second, firms *are* willing to contribute their sensitive loss data into secure platforms such as SCRAM as long as those data are never disclosed to any outside party, including MIT. What this effectively means is that new cryptographic platforms such as SCRAM can gain access to previously ‘untouchable’ data that can then be used to inform market participants and meet important challenges. Many of the target firms for this MPC were interested in participating, but they wanted to see the results of the first computation before contributing their own data. From a cybersecurity standpoint, this represents a new opportunity to create new cybersecurity aggregation pools with greater reach and precision than ever before.

Third, our smaller scale computation confirms the general heavy-tailed distribution of cyber losses reported by other researchers ([Eling & Wirfs, 2019](#); [Leslie et al., 2018](#)), but it also offers new insights into which control failures are leading to those losses. The data highlight a set of problematic controls among large sophisticated firms where security teams may want to focus their own investment decisions and resources. Log management, boundary defenses, asset inventories, and malware defenses appear to be key areas for future emphasis, particularly as this work continues with larger incident sample sizes.

Fourth, although the firms in our sample have a high level of security adoption and sophistication, losses often come from defenses that are well developed, adopted, and understood. In terms of security posture, this is a preliminary indication that improving existing and commonly used defenses should not be neglected in favor of expanding defenses into new areas.

Fifth, we find that the relationship between CIS sub-control adoption and monetary losses is weak in our sample with a correlation of 10.3%. For example, sub-control 12.04 (Deny Communications Over Unauthorized Ports) had the second highest losses of any sub-control, but had 100% adoption across the firms participating in the computation. This is a preliminary indication that the use of checklists may be insufficient for gauging the actual security posture and risk exposure of a firm.

4.2. Implications for Policymakers, Regulators, and Insurers

We hope that this article, and subsequent studies using similar methods, will provide guidance to policymakers, regulators, and insurers, all of whom are trying to establish government rules and private-sector incentives to improve the security of the information systems on which our society increasingly depends. A primary stumbling block in this effort is the lack of any factual basis for setting reasonable expectations for the behavior of institutions, whether public or private. Absent such understanding, it is difficult or impossible to allocate security responsibilities in a clear and economically efficient manner. Our proposed techniques should yield greater insight into the return on investment of different cybersecurity defenses than earlier methods.

With this information, policymakers can use better cyber-risk pricing to write rules on security allocation responsibility in a more informed manner. Regulators with responsibility for sectors such as financial services, health care, utilities, and transportation will be able to fine-tune their supervisory and auditing activities to test firms for behaviors that have been shown to be efficient responses to known risk.

Finally, insurance companies, working with clients and their regulators, will have a better actuarial basis on which to price insurance and empirical guidance about basic industry standards of care. The first steps taken in this article do not yet answer all of these questions, but the methods we have demonstrated, when used at a large scale, should provide the necessary factual background for the institutions responsible for assuring our society is properly protected against cybersecurity risk.

4.3. Implications for the Data Science Community

Our project shows that firms are indeed willing to contribute sensitive data once they are convinced that the process is safe and secure. Cryptographic platforms built around tools such as MPC and SCRAM have the potential to launch a new era of data sharing with privacy and security protections built in. They promise to allow greater access to data sets that were previously too sensitive to disclose and pool, and we expect to see more interesting applications emerge. We built SCRAM to be generalizable and scalable in order to accommodate a broad range of applications and data types. MPC lends itself best to situations where groups of participants would benefit from aggregated statistics and analysis, but are unwilling or unable to disclose their data to any third party.

However, these security and privacy protections do come at a cost in terms of how the data can be used. Data scientists should be aware of the key trade-offs and challenges associated with using these tools.

First, new applications of MPC will likely require additional standardization work and significant preprocessing efforts to ensure that data are submitted without errors and formatted correctly since data can no longer be examined and cleaned once they have been submitted. This means that data preparation at the source takes longer and needs a disproportionate level of attention and education focused on the contributors. In our own computations, we quickly recognized the need for new frameworks to help firms calculate their monetary losses from security incidents.

Second, computations and queries that will be performed on the encrypted data should be discussed and agreed upon by the participants before they are coded into the computation, well before any data are submitted. Data exploration is no longer possible under MPC in the way practitioners use it today. Data scientists will not have the ability to explore the aggregated data, since it always remains encrypted, so any queries or computations must be coded and built directly into the computation ahead of time. In addition, all firms contributing data must agree to decrypt any results coming out of the computation using their own private keys. As the technology matures, it will be important to formalize a governance process among the participants to agree on which computations will be allowed to be coded into the SCRAM platform.

Third, some types of data leakage are still possible if queries or computations only use a narrow set of observations. In Section 2.7, we showed how simple summary statistics can reveal certain elements of the input data when there are too few observations. Solving this challenge involves having a discussion with the participants to get a preliminary sense of what the general data will look like and how many observations the SCRAM platform should expect to receive. Another solution is to run multistage computations that are contingent upon an initial stage confirming the presence of a sufficient number of observations to prevent any data leakage. Additional privacy-preserving techniques, such as differential privacy, could be incorporated into future iterations of SCRAM to address this leakage issue.

Fourth, significant outliers can reveal information about the inputs. This is a particular challenge for cybersecurity, because large incidents are relatively few in number but have extreme values attached to them. For example, our computation had no incident submitted that was larger than \$5 million in losses, but the NotPetya attack is estimated to have cost individual firms over 100 times that amount ([Greenburg, 2018](#)). This would have left the contributing factors assigned by a company with a large outlier visible because of the sheer size of the loss relative to the others. As discussed in Section 2.8, there are ways to screen for outliers, either through preprocessing at the client before data are encrypted and submitted, or via a multistage computation. These need to be planned ahead of time and built into the software. Ultimately, however, larger sample sizes will ameliorate the problem.

4.4. Future Work

These results provide a compelling proof-of-concept for how cyber-intrusion data can be shared. Our next step will be to increase the number of participants and, consequently, incidents in future rounds to produce more robust estimates, more complex analyses, and more generalizable results. With a larger sample, we will also be able to explore loss distribution approaches that cover both the frequency and severity of losses ([Eling & Wirfs, 2019](#); [Panjer, 2006](#)). More data will also reduce the chance of outliers or single incidents leaking the magnitude of an individual event.

Further work is needed to help firms standardize how they record their cyber losses. Standardization has an oversized role in MPC implementations since the input data are never revealed (see Sections 2.6 and 4.3). Our research uncovered the need for greater standardization in calculating monetary losses of cyber incidents. This will be necessary to expand the number of participating firms. In general, future work should focus on standardizing data collection about attacks, monetary losses, and other damages for further use on the platform.

This initial research gathered firms across sectors to contribute data, but future work with more firms could produce both economy-wide and sector-specific computations to compare defense postures, attacks, and monetary losses. Sector-specific computations will require sample sizes that are large enough to mask the inputs of firms in each sector, particularly when there are outliers.

Our research focused on financial losses from cyber incidents because they are the most straightforward way to compare harm across firms, but our participants highlighted other significant impacts, such as reputational damage and intellectual property theft. These consequences may not yield large and immediate monetary losses, but may have a significant impact on future revenues or other critical aspects of the firm. Future research should examine other types of cyber harm ([Agrafiotis et al., 2018](#)).

Acknowledgments

This research was funded by MIT’s Internet Policy Research Initiative, the MIT Laboratory for Financial Engineering, and FinTech@CSAIL. We thank the anonymous reviewers and the editor for their insightful comments and recommendations throughout the review process that helped us improve the article. We are very grateful to the companies who contributed their security data and sent representatives to MIT to run the computation. We thank Ari Schwartz at Venable and the Cybersecurity Policy and Law Coalition he leads for helping us formulate the issues. We also thank Jayna Cummings for her project expertise bringing together the participants and for editorial assistance, Jeff Schiller for his help on code development, Matthew Briggs for setting up the in-person computation in 2019, and Catherine Fairclough for editing assistance.

Disclosure Statement

Vinod Vaikuntanathan is a co-founder of a commercial venture (Duality Technologies) that sells privacy-protecting computing products to various industries. The research in this article is independent and was funded and developed separately from the company. There has been no contact or code sharing with Duality. Daniel J. Weitzner is a member of the Scientific Advisory Board of a commercial venture (Duality Technologies) that sells privacy-protecting computing products to various industries. The research in this article is independent and was funded and developed separately from the company. There has been no contact or code sharing with the commercial venture. The other coauthors report no conflicts.

Appendix

Cryptographic Details

In this appendix, we give a more detailed description of the cryptographic algorithms used in our MPC SCRAM platform. The intended audience of this section is a reader with a background in lattice cryptography and familiarity with the Ring-Learning with Errors (RLWE) problem. Familiarity with the homomorphic encryption scheme of ([Brakerski, 2012](#)) and ([Fan & Vercauteren, n.d.](#)) would also be helpful. Familiarity with ([Asharov et al., 2012](#)) is not necessary to understand these algorithms.

The main cryptographic primitive used for SCRAM was homomorphic encryption, specifically the scheme of ([Brakerski, 2012](#)) and ([Fan & Vercauteren, n.d.](#)), from here on referred to as the BFV scheme. Homomorphic

operations are defined in this scheme for ciphertexts decryptable with the same secret key. Our approach to designing an MPC algorithm based on the BFV scheme was to design a key generation protocol and corresponding decryption protocol following the work of (Asharov et al., 2012), where all parties participating in the MPC protocol could encrypt their data with the same public key, but a ciphertext would only be able to be decrypted if authorization is given by all parties. The algorithms described below provide more detail of our specific instantiation of the framework of (Asharov et al., 2012).

A.1. Distributed Key Generation

Consider the standard BFV key generation protocol, which outputs a public-key/secret-key pair of the form

$$(\mathbf{pk}, \mathbf{sk}) = ((\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + \mathbf{e}), \mathbf{s})$$

where $\mathbf{a}$ is a uniformly random sample over the ring $\mathcal{R}_q = \mathbb{Z}_q[x]/f(x)$, where $f(x)$ is some degree- n polynomial. Our key generation protocol relies on the common random string (CRS) model and begins with the assumption that all participants have access to the same truly random seed for a pseudorandom number generator (PRNG). All participants then use this seed to generate the same pseudorandom sample $\mathbf{a}$ from $\mathcal{R}_q$. Each party i then generates the following key pair:

$$(\mathbf{pk}_i, \mathbf{sk}_i) = ((\mathbf{a}, \mathbf{a} \cdot \mathbf{s}_i + \mathbf{e}_i), \mathbf{s}_i).$$

Once all these public keys are generated, the shared public key is computed by summing the second component of each $\mathbf{pk}_i = (\mathbf{a}, \mathbf{pk}_i[1])$ over $\mathcal{R}_q$ to get the following:

$$\mathbf{pk} = (\mathbf{a}, \sum_i \mathbf{pk}_i[1]) = (\mathbf{a}, \mathbf{a} \cdot \sum_i \mathbf{s}_i + \sum_i \mathbf{e}_i).$$

The result of this sum is a well-formed public key for the BFV scheme, where the corresponding secret key, $\mathbf{sk} = \sum_i \mathbf{s}_i$, is already distributed in additive shares among the participants. While the error term grows by a factor of k , where k is the number of parties, we are only considering the case where the number of parties is small. For a larger number of parties, the parameters of the scheme can be easily adjusted to accommodate this slightly increased error term.

In the next section, we describe how decryption is performed on a ciphertext encrypted with this public key.

A.2. Distributed Decryption

Consider a ciphertext in the standard BFV scheme, which has the following form:

$$\mathbf{ct} = (\mathbf{a}, \mathbf{a} \cdot \mathbf{s} + \Delta \mathbf{m} + \mathbf{e})$$

where Δ is a public scaling factor to prevent the small error term, $\mathbf{e}$, from corrupting the message, $\mathbf{m}$. Decryption is performed by computing the following function on $\mathbf{ct} = (\mathbf{ct}[0], \mathbf{ct}[1])$.

$$\mathbf{m} = \left\lfloor \frac{\mathbf{ct}[1] - \mathbf{s} \cdot \mathbf{ct}[0]}{\Delta} \right\rfloor = \left\lfloor \frac{\Delta \mathbf{m} + \mathbf{e}}{\Delta} \right\rfloor.$$

Our distributed decryption protocol collaboratively computes the numerator from “decryption shares” generated by each party, then the division and flooring is computed in the clear. Recall from the previous section that the secret key has the form $s = \sum_{i=1}^k s_i$, and each party i has a share s_i of the secret key. We assume that all parties know the total number of parties, k , as well as the ciphertext, $ct = (a, a \cdot s + \Delta m + e)$, that is to be decrypted. Each client i takes their share of the secret key, s_i , and generates the following decryption share:

$$d_i = a \cdot s + \Delta m + e - k \cdot (a \cdot s_i + e'_i)$$

where all operations are over $\mathcal{R}_q$ and e'_i is an error term sampled from a discrete Gaussian with large standard deviation.³ The additive term $(a \cdot s_i + e'_i)$ is multiplied by k because when these shares are summed there are k terms with $a \cdot s = a \sum_i s_i$, each with a component $a \cdot s_i$ that must be subtracted away.

Each sample d_i is then published. The security of this step follows from the fact that the tuple $(a, a \cdot s_i + e'_i)$ is a well-formed RLWE sample, which is computationally indistinguishable from (a, u) for a uniformly sampled $u \leftarrow \mathcal{R}_q$.

When all of the decryption shares have been published, the numerator in the decryption equation can be computed as follows:

$$\begin{aligned} d &= \sum_{i=1}^k d_i = k \cdot a \cdot s + k \cdot \Delta m + k \cdot e - \sum_{i=1}^k (k \cdot a \cdot s_i + e'_i) \\ &= k \cdot a \cdot s + k \cdot \Delta m + k \cdot e - k \cdot a \sum_{i=1}^k (s_i) - e' \\ &= k \cdot a \cdot s + k \cdot \Delta m + k \cdot e - k \cdot a \cdot s - e' \\ &= k \cdot \Delta m + k \cdot e - e' \end{aligned}$$

Note that the message, m , is multiplied by k in this numerator.

Once this numerator is computed, the message, m , can be recovered by simply dividing by $k\Delta$ and flooring.

$$m = \left\lfloor \frac{d}{k\Delta} \right\rfloor = \left\lfloor \frac{k \cdot \Delta m + k \cdot e - e'}{k\Delta} \right\rfloor.$$

Correctness holds as long as the magnitude of the error term, $ke - e'$, remains less than $k\Delta/2$.

Using these two protocols, we have an additive homomorphic encryption scheme that allows us to safely encrypt data from multiple parties with the same BFV public key. This, in turn, allows us to use additive homomorphic operations on the encrypted data to compute the desired function of the MPC protocol, and then use the above distributed decryption protocol to allow the parties to obtain the function output.

A.3. Parameter Selection

We chose parameters for our scheme based on the NIST proposal to standardize the security of homomorphic encryption schemes (Albrecht et al., 2018). Based on table 1 in (Albrecht et al., 2018), our parameters give 128 bits of security. Since our RLWE secrets were drawn from the error distribution, we chose the 128-bit security level parameters from this table, which gave us a degree $n = 8192$ and a maximum ciphertext modulus of 2^{220} . Our ciphertext modulus was the product of three 55-bit primes, which gives a ciphertext modulus of 165 bits.

References

- Abbe, E. A., Khandani, A. E., & Lo, A. W. (2012). Privacy-preserving methods for sharing financial risk exposures. *American Economic Review*, 102(3), 65–70. <https://doi.org/10.1257/aer.102.3.65>
- Abidin, A., Aly, A., Cleemput, S., & Mustafa, M. A. (2016). An mpc-based privacy-preserving protocol for a local electricity trading market. In S. Foresti & G. Persiano (Eds.), *Cryptology and network security* (pp. 615–625). Springer International Publishing.
- Advisen. (2019). *Cyber loss data*. <https://www.advisenltd.com/data/cyber-loss-data/>
- Agrafiotis, I., Nurse, J. R., Goldsmith, M., Creese, S., & Upton, D. (2018). A taxonomy of cyber-harms: Defining the impacts of cyber-attacks and understanding how they propagate. *Journal of Cybersecurity*, 4(1), Article ty006. <https://doi.org/10.1093/cybsec/tyy006>
- Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., Lokam, S., Micciancio, D., Moody, D., Morrison, T., Sahai, A., & Vaikuntanathan, V. (2018). *Homomorphic encryption security standard*. Homomorphic Encryption. <http://homomorphicecryption.org/wp-content/uploads/2018/11/HomomorphicEncryptionStandardv1.1.pdf>
- Anderson, R., Barton, C., Böhme, R., Clayton, R., Ganán, C., Grasso, T., Levi, M., Moore, T., & Vasek, M. (2019). Measuring the changing cost of cybercrime. In *the 18th Annual Workshop on the Economics of Information Security*. <https://doi.org/10.17863/CAM.41598>
- Anderson, R., Barton, C., Böhme, R., Clayton, R., Van Eeten, M. J., Levi, M., Moore, T., & Savage, S. (2013). Measuring the cost of cybercrime. In *The economics of information security and privacy* (pp. 265–300). Springer. <https://doi.org/10.17863/CAM.41598>
- Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., & Wichs, D. (2012). Multiparty computation with low communication, computation and interaction via threshold FHE. In D. Pointcheval & T. Johansson (Eds.), *Lecture Notes in Computer Science: Vol. 7237. Advances in Cryptology – EUROCRYPT 2012* (pp. 483–501). Springer. https://doi.org/10.1007/978-3-642-29011-4_29

Bandyopadhyay, T., Mookerjee, V. S., & Rao, R. C. (2009). Why it managers don't go for cyber-insurance products. *Communications of the ACM-Scratch Programming for All*, 52(11), 68–73.

<https://doi.org/10.1145/1592761.1592780>

Bogdanov, D., Kamm, L., Kubo, B., Rebane, R., Sokk, V., & Talviste, R. (2016). Students and taxes: A privacy-preserving study using secure computation. *Proceedings on Privacy Enhancing Technologies*, 2016(3), 117–135. <https://doi.org/10.1515/popets-2016-0019>

Bogetoft, P., Christensen, D. L., Damgård, I., Geisler, M., Jakobsen, T., Krøigaard, M., Nielsen, J. D., Nielsen, J. B., Nielsen, K., & Pagter, J. (2009). *Secure multiparty computation goes live*. In R. Dingledine & P. Golle (Eds.), *Lecture Notes in Computer Science: Vol. 5628. FC 2009: Financial Cryptography and Data Security* (pp. 325–343). https://doi.org/10.1007/978-3-642-03549-4_20

Boston University (n.d.). *Accessible and scalable secure multi-party computation*. <https://multiparty.org/>

Bowne-Anderson, H. (2018). What data scientists really do, according to 35 data scientists. *Harvard Business Review*, 8, 45–57. <https://hbr.org/2018/08/what-data-scientists-really-do-according-to-35-data-scientists>

Brakerski, Z. (2012). Fully homomorphic encryption without modulus switching from classical GapSVP. In R. Safavi-Naini & R. Canetti (Eds.), *Lecture Notes in Computer Science: Vol. 7417. CRYPTO 2012: Advances in Cryptology – CRYPTO 2012* (pp 868–886).

Brenner, J. (2017). *Keeping america safe: Toward more secure networks for critical sectors*. Massachusetts Institute of Technology, Internet Policy Research Initiative.

Cavusoglu, H., Mishra, B., & Raghunathan, S. (2004). The effect of internet security breach announcements on market value: Capital market reactions for breached firms and internet security developers. *International Journal of Electronic Commerce*, 9(1), 70–104. <https://doi.org/10.1080/10864415.2004.11044320>

Corrigan-Gibbs, H., & Boneh, D. (2017). Prio: Private, robust, and scalable computation of aggregate statistics. In *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)* (pp. 259–282). <https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/corrigan-gibbs>

Eling, M., & Wirfs, J. (2019). What are the actual costs of cyber risk events? *European Journal of Operational Research*, 272(3), 1109–1119. <https://doi.org/10.1016/j.ejor.2018.07.021>

Fan, J., & Vercauteren, F. (n.d.). *Somewhat practical fully homomorphic encryption*. ia.cr/2012/144

Geer, D., Hoo, K. S., & Jaquith, A. (2003). Information security: Why the future belongs to the quants. *IEEE Security & Privacy*, 1(4), 24–32. <https://doi.org/10.1109/MSECP.2003.1219053>

- Goldreich, O., Micali, S., & Wigderson, A. (1987). How to play any mental game. In B. Shriver (Ed.), *Proceedings of the nineteenth annual ACM symposium on theory of computing* (pp. 218–229). Association for Computing Machinery. <https://doi.org/10.1145/28395.28420>
- Goldreich, O., Micali, S., & Wigderson, A. (1991). Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *Journal of the ACM*, 38(3), 690–728. <https://doi.org/10.1145/116825.116852>
- Greenburg, A. (2018, August 22). The untold story of notpetya, the most devastating cyberattack in history. *Wired*. <https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/>
- Hofmann, A., Wheatley, S., & Sornette, D. (2019). Heavy-tailed data breaches in the nat-cat framework & the challenge of insuring cyber risks. *arXiv*. <https://doi.org/10.48550/arXiv.1901.00699>
- Hubbard, D. W., & Seiersen, R. (2016). *How to measure anything in cybersecurity risk*. John Wiley & Sons, Inc. <https://doi.org/10.1002/9781119162315>
- Ion, M., Kreuter, B., Nergiz, A. E., Patel, S., Raykova, M., Saxena, S., Seth, K., Shanahan, D., & Yung, M. (2019). On deploying secure computing commercially: Private intersection-sum protocols and their business applications. *IACR Cryptology EPrint Archive*, 2019, 723. <https://eprint.iacr.org/2019/723>
- Kamm, L., Bogdanov, D., Laur, S., & Vilo, J. (2013). A new way to protect privacy in large-scale genome-wide association studies. *Bioinformatics*, 29(7), 886–893. <https://doi.org/10.1093/bioinformatics/btt066>
- Lapets, A., Volgushev, N., Bestavros, A., Jansen, F., & Varia, M. (2016). Secure mpc for analytics as a web application. *2016 IEEE Cybersecurity Development (SecDev)*, 73–74. <https://doi.org/10.1109/SecDev.2016.027>
- Leslie, N. O., Harang, R. E., Knachel, L. P., & Kott, A. (2018). Statistical models for the number of successful cyber intrusions. *The Journal of Defense Modeling and Simulation*, 15(1), 49–63. <https://doi.org/10.1177/1548512917715342>
- Microsoft. (2020). *Microsoft seal (release 3.5)*. GitHub. <https://github.com/Microsoft/SEAL>
- NetDiligence. (2018). *2018 cyber claims study*.
- Operational Riskdata exchange association (ORX). (2017). *Beyond the headlines: Insurance - operational risk loss data for insurers*. <https://managingrisktogether.orx.org/orx-membership/loss-data>
- Panjer, H. H. (2006). *Operational risk: Modeling analytics* (Vol. 620). John Wiley & Sons.
- Polyakov, Y., Rohloff, K., & Ryan, G. W. (n.d.). *PALISADE lattice cryptography library*. GitHub. <https://git.njit.edu/palisade/PALISADE>

- Ponemon Institute. (2016). *Cost of cyber crime study*. <https://www.ponemon.org/local/upload/file / 2016%5C%20HPE% 5C%20CCC %5C% 20GLOBAL% 5C %20REPORT% 5C%20FINAL%5C%203.pdf>
- Powers, M. R. (2007). Using aumann-shapley values to allocate insurance risk: The case of inhomogeneous losses. *North American Actuarial Journal*, 11(3), 113–127. <https://doi.org/10.1080/10920277.2007.10597470>
- SAS Corporation. (2015). *OpRisk global data*. https://www.sas.com/content/dam/SAS/en_us/doc/productbrief/sas-oprisk-global-data-101187.pdf
- Swiss Re Institute (2017). *Cyber: Getting to grips with a complex risk*. https://www.swissre.com/dam/jcr:995517ee-27cd-4aae-b4b1-44fb862af25e/sigma1_2017_en.pdf
- U. S. Council of Economic Advisers (2018). *The cost of malicious cyber activity to the U.S. economy*. <https://www.whitehouse.gov/wp-content/uploads/2018/03/The-Cost-of-Malicious-Cyber-Activity-to-the-U.S.-Economy.pdf>
- Verizon Corporation. (2017). *Vocabulary for event recording and incident sharing (veris)*. <http://vcdb.org/>
- Wheatley, S., Maillart, T., & Sornette, D. (2016). The extreme risk of personal data breaches and the erosion of privacy. *The European Physical Journal B*, 89(1), Article 7. <https://doi.org/10.1140/epjb/e2015-60754-4>
- Yao, A. C.-C. (1986). How to generate and exchange secrets. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science* (pp. 162–167). <https://doi.org/10.1109/SFCS.1986.25>

©2020 Leo de Castro, Andrew W. Lo, Taylor Reynolds, Fransisca Susan, Vinod Vaikuntanathan, Daniel Weitzner, and Nicolas Zhang. This article is licensed under a Creative Commons Attribution (CC BY 4.0) [International license](#), except where otherwise indicated with respect to particular material included in the article.

Footnotes

1. The CIS numbering scheme uses the format (control.subcontrol). We implemented a slightly modified numbering scheme for CIS sub-controls throughout this article to aid with readability and data analysis. In our numbering, we add a leading zero after the decimal point to single-digit sub-controls; thus 6.4 (CIS original) becomes 6.04 (our modification), and 18.1 (CIS original) becomes 18.01 (our modification). Double-digit sub-controls such as 12.10 stay the same in both numbering schemes. [↵](#)
2. Seven firms contributed data on our first attempt, but the results came back with clearly incorrect output that could only have been caused by a formatting error. In one-on-one discussions with each of the participants, we were able to determine which contributor had the formatting issue and excluded that firm

before running the computation again. The results in our article are limited to the successful run with six firms. ↵

3. Such noise terms are sometimes referred to as 'flooding' or 'smudging' terms. ↵

References

- Abbe, E. A., Khandani, A. E., & Lo, A. W. (2012). Privacy-preserving methods for sharing financial risk exposures. *American Economic Review*, 102(3), 65–70. <https://doi.org/10.1257/aer.102.3.65> ↵
- Abidin, A., Aly, A., Cleemput, S., & Mustafa, M. A. (2016). An mpc-based privacy-preserving protocol for a local electricity trading market. In S. Foresti & G. Persiano (Eds.), *Cryptology and network security* (pp. 615–625). Springer International Publishing. ↵
- Advisen. (2019). Cyber Loss Data. <https://www.advisenltd.com/data/cyber-loss-data/> ↵
- Agrafiotis, I., Nurse, J. R., Goldsmith, M., Creese, S., & Upton, D. (2018). A taxonomy of cyber-harms: Defining the impacts of cyber-attacks and understanding how they propagate. *Journal of Cybersecurity*, 4(1), ty006. <https://doi.org/10.1093/cybsec/tyy006> ↵
- Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., Lokam, S., Micciancio, D., Moody, D., Morrison, T., Sahai, A., & Vaikuntanathan, V. (2018). *Homomorphic encryption security standard* [Techreport]. ↵
- Anderson, R., Barton, C., Böhme, R., Clayton, R., Ganán, C., Grasso, T., Levi, M., Moore, T., & Vasek, M. (2019). Measuring the changing cost of cybercrime. <https://doi.org/10.17863/CAM.41598> ↵
- Anderson, R., Barton, C., Böhme, R., Clayton, R., Van Eeten, M. J., Levi, M., Moore, T., & Savage, S. (2013). Measuring the cost of cybercrime. In *The economics of information security and privacy* (pp. 265–300). Springer. <https://doi.org/10.17863/CAM.41598> ↵
- Asharov, G., Jain, A., López-Alt, A., Tromer, E., Vaikuntanathan, V., & Wichs, D. (2012). *Multiparty computation with low communication, computation and interaction via threshold FHE*. In D. Pointcheval & T. Johansson (Eds.), *Advances in cryptology - EUROCRYPT 2012 - 31st annual international conference on the theory and applications of cryptographic techniques, cambridge, uk, april 15-19, 2012. Proceedings* (Vol. 7237, pp. 483–501). Springer. <https://doi.org/10.1007/978-3-642-29011-4\textbackslash} 29> ↵
- Bandyopadhyay, T., Mookerjee, V. S., & Rao, R. C. (2009). Why it managers don't go for cyber-insurance products. *Communications of the ACM-Scratch Programming for All*, 52(11), 68. <https://doi.org/10.1145/1592761.1592780> ↵
- Bogdanov, D., Kamm, L., Kubo, B., Rebane, R., Sokk, V., & Talviste, R. (2016). Students and taxes: A privacy-preserving study using secure computation. *Proceedings on Privacy Enhancing Technologies*, 2016(3), 117–135. <https://doi.org/10.1515/popets-2016-0019> ↵
- Bogetoft, P., Christensen, D. L., Damgård, I., Geisler, M., Jakobsen, T., Krøigaard, M., Nielsen, J. D., Nielsen, J. B., Nielsen, K., & Pagter, J. (2009). *Secure multiparty computation goes live*. 325–343. <https://doi.org/10.1007/978-3-642-03549-4 20> ↵

- Bowne-Anderson, H. (2018). What data scientists really do, according to 35 data scientists. *Harvard Business Review*, 8, 45–57. <https://hbr.org/2018/08/what-data-scientists-really-do-according-to-35-data-scientists> ↵
- Brakerski, Z. (2012). Fully homomorphic encryption without modulus switching from classical gapsvp. *Advances in Cryptology—CRYPTO 2012*, 868–886. <https://doi.org/10.1007/978-3-642-32009-5> ↵
- Brenner, J. (2017). Keeping america safe: Toward more secure networks for critical sectors. Massachusetts Institute of Technology, Internet Policy Research Initiative. ↵
- Cavusoglu, H., Mishra, B., & Raghunathan, S. (2004). The effect of internet security breach announcements on market value: Capital market reactions for breached firms and internet security developers. *International Journal of Electronic Commerce*, 9(1), 70–104. <https://doi.org/10.1080/10864415.2004.11044320> ↵
- Corporation, S. (2015). *OpRisk Global Data*. https://www.sas.com/content/dam/SAS/en_us/doc/productbrief/sas-oprisk-global-data-101187.pdf ↵
- Corporation, V. (2017). *Vocabulary for event recording and incident sharing (veris)*. <http://vcdb.org/> ↵
- Corrigan-Gibbs, H., & Boneh, D. (2017). Prio: Private, robust, and scalable computation of aggregate statistics. *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, 259–282. <https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/corrigan-gibbs> ↵
- Eling, M., & Wirfs, J. (2019). What are the actual costs of cyber risk events? *European Journal of Operational Research*, 272(3), 1109–1119. <https://doi.org/10.1016/j.ejor.2018.07.021> ↵
- exchange association (ORX), O. R. (2017). *Beyond the headlines: Insurance - operational risk loss data for insurers*. <https://managingrisktogether.orx.org/orx-membership/loss-data> ↵
- Fan, J., & Vercauteren, F. (n.d.). *Somewhat practical fully homomorphic encryption*. ia.cr/2012/144 ↵
- Geer, D., Hoo, K. S., & Jaquith, A. (2003). Information security: Why the future belongs to the quants. *IEEE Security & Privacy*, 1(4), 24–32. <https://doi.org/10.1109/MSECP.2003.1219053> ↵
- Goldreich, O., Micali, S., & Wigderson, A. (1987). How to play any mental game. In B. Shriver (Ed.), *Proceedings of the nineteenth annual acm symposium on theory of computing* (pp. 218–229). Association for Computing Machinery. <https://doi.org/10.1145/28395.28420> ↵
- Goldreich, O., Micali, S., & Wigderson, A. (1991). Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3), 690–728. <https://doi.org/10.1145/116825.116852> ↵
- Greenburg, A. (2018). The untold story of notpetya, the most devastating cyberattack in history ([Wired.com](https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/), Ed.). [<https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/>] (<https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/>). <https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/%20> ↵
- Hofmann, A., Wheatley, S., & Sornette, D. (2019). Heavy-tailed data breaches in the nat-cat framework & the challenge of insuring cyber risks. *arXiv Preprint arXiv:1901.00699*. ↵

- Hubbard, D. W., & Seiersen, R. (2016). *How to measure anything in cybersecurity risk*. <https://doi.org/10.1002/9781119162315> ↵
- Institute, P. (2016). *Cost of cyber crime study*. <https://www.ponemon.org/local/upload/file/2016/{\textbackslash}%20HPE{\textbackslash}%20CCC{\textbackslash}%20GLOBAL{\textbackslash}%20REPORT{\textbackslash}%20FINAL{\textbackslash}%203.pdf> ↵
- Institute, S. R. (2017). Cyber: Getting to grips with a complex risk. https://www.swissre.com/dam/jcr:995517ee-27cd-4aae-b4b1-44fb862af25e/sigma1_2017_en.pdf ↵
- Ion, M., Kreuter, B., Nergiz, A. E., Patel, S., Raykova, M., Saxena, S., Seth, K., Shanahan, D., & Yung, M. (2019). On deploying secure computing commercially: Private intersection-sum protocols and their business applications. *IACR Cryptology ePrint Archive*, 2019, 723. <https://eprint.iacr.org/2019/723> ↵
- Kamm, L., Bogdanov, D., Laur, S., & Vilo, J. (2013). A new way to protect privacy in large-scale genome-wide association studies. *Bioinformatics*, 29(7), 886–893. <https://doi.org/10.1093/bioinformatics/btt066> ↵
- Lapets, A., Volgushev, N., Bestavros, A., Jansen, F., & Varia, M. (2016). Secure mpc for analytics as a web application. *2016 IEEE Cybersecurity Development (SecDev)*, 73–74. <https://doi.org/10.1109/SecDev.2016.027> ↵
- Leslie, N. O., Harang, R. E., Knachel, L. P., & Kott, A. (2018). Statistical models for the number of successful cyber intrusions. *The Journal of Defense Modeling and Simulation*, 15(1), 49–63. <https://doi.org/10.1177/1548512917715342> ↵
- Microsoft. (2020). *Microsoft seal (release 3.5)*. <<https://github.com/Microsoft/SEAL>>:; <https://github.com/Microsoft/SEAL> ↵
- NetDiligence. (2018). 2018 cyber claims study. ↵
- of Economic Advisers, U. S. C. (2018). The cost of malicious cyber activity to the u.s. Economy. <https://www.whitehouse.gov/wp-content/uploads/2018/03/The-Cost-of-Malicious-Cyber-Activity-to-the-U.S.-Economy.pdf> ↵
- Panjer, H. H. (2006). *Operational risk.: modeling analytics* (Vol. 620). John Wiley & Sons. ↵
- Polyakov, Y., Rohloff, K., & Ryan, G. W. (n.d.). PALISADE lattice cryptography library. <<https://git.njit.edu/palisade/PALISADE>>:; ↵
- Powers, M. R. (2007). Using aumann-shapley values to allocate insurance risk: The case of inhomogeneous losses. *North American Actuarial Journal*, 11(3), 113–127. ↵
- University, B. (n.d.). Accessible and scalable secure multi-party computation. <https://multiparty.org/> ↵
- Wheatley, S., Maillart, T., & Sornette, D. (2016). The extreme risk of personal data breaches and the erosion of privacy. *The European Physical Journal B*, 89(1), 7. <https://doi.org/10.1140/epjb/e2015-60754-4> ↵
- Yao, A. C.-C. (1986). How to generate and exchange secrets. *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, 162–167. <https://doi.org/10.1109/SFCS.1986.25> ↵