Multi-Party Computation for Contemporary Web Services and Data Workflows

Andrei Lapets
andrei@nthparty.com

Frederick Jansen
frederick@nthparty.com

Shereen Shermak
shereen@nthparty.com

Abstract

By building and deploying web services or data analysis workflows that employ secure multi-party computation (MPC), organizations may be able to provide new services, to identify and leverage new business opportunities, and to reduce risks for both themselves and their customers. MPC is already being incorporated into software solutions in some domains. However, using MPC still requires building an understanding of how its features can satisfy technical, business, and legal constraints, as well as some custom engineering effort. In this document, we introduce MPC and its security and privacy advantages, review some context around MPC and related cryptographic techniques, and provide an overview of the current landscape of MPC capabilities and opportunities, with particular emphasis on scenarios and challenges for which MPC is ready today.

1 Introduction

Secure multi-party computation (MPC) is a family of cryptographic techniques that allow organizations and individuals to enjoy the benefits of web-based services and data analysis workflows while mitigating or removing the risks traditionally associated with providing or sharing the data that those services and workflows require. MPC can be used both to reduce risks associated with existing workflows and to enable new opportunities in scenarios in which data sharing is encumbered or restricted by the security and privacy concerns of individuals, the policies of companies and other organizations, and legal constraints and regulations.

Nth Party builds and provides products and expertise that help organizations introduce MPC into their web services and data analysis workflows. Relying on years of experience developing and deploying MPC applications [9, 14], and building on open-source frameworks and software that have been proven in real-world deployments [1], the Nth Party team has assembled a rich suite of libraries and tools that can be used to quickly assemble MPC solutions to address contemporary privacy and security challenges.

2 Security and Privacy Advantages of MPC

In traditional services and workflows that rely on computation over sensitive or private data that is encrypted, it is usually necessary to decrypt that data at some location and for some period of time so that existing computational tools can be applied to it (*e.g.*, within a *data clean room* that is set up temporarily so that two organizations can run analyses on their joint datasets inside it). The result of the computation might then be encrypted again before it leaves the location where the computation took place. Novel techniques such as MPC make it possible to remove this requirement: sensitive or private data never needs to be decrypted at any point in the process. This means that the risks, liabilities, and costs associated with protecting the data while it is in a decrypted form can be reduced or eliminated.

2.1 Common Features and Variants

How computation over data that is not in a decrypted form in a particular location is made possible depends on the particular MPC technique that is involved. However, the common characteristic is that at least two parties must be involved to make MPC possible.

2.1.1 Random Values

The security properties of MPC rely in part on the ability of participating parties to generate random values privately (*i.e.*, the process they use to generate the random value cannot be observed or influenced by other parties). To understand how this capability can be leveraged to facilitate secure information exchange, suppose we observe that an output of a computation is 9, and we know that this output was obtained from some input s by adding the value 7 to it. We can then solve the following equation to conclude that $s = 2$:

$$s + 7 = 9$$

Now, suppose that instead, all we know is that the output 9 was obtained by choosing a value r uniformly at random (*i.e.*, every possibility is equally likely) in the range $\{0, 1, 2, 3, \ldots, 9\}$ and adding it to s:

$$s + r = 9$$

At this point, we can say that given our observation of 9, any value in the range $\{0, 1, 2, 3, \ldots, 9\}$ is *equally likely* to have been the original value of s. In other words, our observation of $s = 9$ and our knowledge about how r was chosen leads us to conclude that we *must* be in one of the following equally likely scenarios:

$$\begin{array}{ccccc} 0+9=9 & 1+8=9 & 2+7=9 & 3+6=9 & 4+5=9 \\ 5+4=9 & 6+3=9 & 7+2=9 & 8+1=9 & 9+0=9 \end{array}$$

If $\{0, 1, 2, 3, \ldots, 9\}$ represents the entire range of possible values of s in our scenario, then observing the output 9 gives us no information about s.

The security properties of MPC techniques rely on the storage and exchange of information that appears random in the manner described above. Usually, the actual ranges of values are drawn from a cyclical mathematical structure in which operations “wrap around”, such as a group, so that no upper or lower bounds can be inferred from the observation of a random value [5].

2.1.2 Secret Sharing

A common building block of MPC is *secret sharing* [15], a well-known cryptographic primitive (among other primitives such as hash functions and public-key encryption functions) that has strong links to secure computation. When sensitive data is in secret-shared form, it is split across two or more parties in such a way that the information available to any individual party consists of values chosen uniformly at random from a fixed range.

While in a secret-shared state, data can be understood as being encrypted (*i.e.,*, in its ciphertext form). For example, consider a sensitive value 16 when it is secret-shared between Party A and Party B as two values: -7 is held by Party A and 23 is held by Party B. We can think of -7 as an encryption key that is in the possession of Party A, and 23 as the encrypted value or ciphertext that is in the possession of Party B (or vice versa). Assuming that the parties involved do not exchange the values in their possession with one another, the security properties of MPC protocols that employ secret sharing can be understood to be equivalent to those of the one-time pad, which is an encryption technique that provides information-theoretic security [16].

For any implementation that involves multiple parties it is important to communicate clearly which parties possess secret shares and under what circumstances those secret shares can be recombined, as this affects the risks associated with distributing secret shares. Suppose Party A splits its data s into two secret shares b and c such that $a = b + c$. If Party A delivers b to Party B and c to Party C, then there is a risk that Party B and Party C may recombine their shares without the consent of Party A. On the other hand, if Party A delivers b to Party B but keeps c to itself, then a is also in secret-shared form. But in this case, the risk that Party B may obtain the other share c that it needs to reconstruct a is *equivalent* to the risk that Party B may obtain the original value a. Thus, it can be the case that secret sharing data while retaining one of the secret shares introduces no additional risks to a data owner beyond possession of the original data itself.

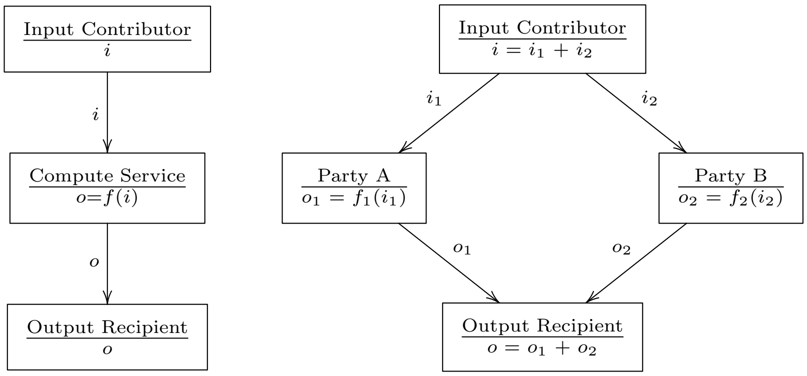

Figure 1: Computation on data compared to MPC over secret-shared data. In the MPC scenario on the right, the contributor chooses secret shares i_1 and i_2 in a random way as in Section 2.1.2.

Secret sharing can be compatible with computation. As illustrated in Figure 1, each party may perform operations on the shares in its own possession. In the case of additive secret sharing (which supports only addition operations), the parties need not communicate at all until it is time to reconstruct the final output [5, 12]. In more general MPC techniques, the parties may also need to communicate with one another during the computation. However, just as in the case of secret sharing, the values received and observed by each participating party would appear to be chosen uniformly at random from a fixed range (from the perspective of that party).

2.1.3 Other MPC Techniques

Other MPC techniques such as garbled circuits [3], oblivious transfer [8], and oblivious pseudorandom functions [10] can be used to enable the computation of functions and algorithms over data from multiple contributors in a manner that does not require the data to be shared among those participants. These techniques can be viewed as alternatives, extensions, or applications of secret sharing that have their own distinct compatibility, performance, and security characteristics. However, all of the communications protocols involved in these techniques rely on the exchange of information that appears random to recipients.

2.1.4 Cryptographic Primitives and Key Management

Some MPC techniques rely on well-known and ubiquitous cryptographic primitives such as symmetric encryption, public-key encryption, and hashing. Some protocols use these primitives in a way that insulates participants from issues such as key management, often because these primitives are used at a much lower level to provide more high-level functionalities (this is among the reasons that MPC protocols sometimes have significant performance and communications overheads). In these cases, organizations that deploy MPC technologies may find that they need not devote any additional resources to cryptographic key management processes beyond those that meet their existing organizational requirements. In other protocols and deployment scenarios, participants may need to do some key storage and management [14]. In such scenarios, it would be especially important to ensure that these tasks are handled according to internal and industry best practices, as well as in compliance with applicable regulations.

2.1.5 Security and Different Categories of Adversarial Behavior

Because multiple parties must cooperate in some defined way for MPC to function, the security benefits of each technique rely on assumptions about how the parties involved may or may not choose to misbehave throughout the process. Two of the most commonly studied forms of adversarial behavior on the part of MPC participants are *honest-but-curious* and *malicious*. Honest-but-curious participants will run MPC software as prescribed but may attempt to glean information along the way. Malicious participants will perform any action necessary to breach the confidentiality or integrity properties of the protocol. When choosing an MPC protocol for a particular scenario, it is important to consider into which of these two categories participants may fall. Some protocols only provide security against honest-but-curious adversaries, while others can be tuned for either type of adversarial behavior. It is often the case that security against malicious adversaries will require a protocol that has more significant communications and performance overheads.

The security benefits of MPC rely on the fact that parties will not *collude* by exchanging their respective secret shares in inappropriate ways (a type of malicious behavior). Thus, another consideration when choosing a protocol is how many parties must collude in order to break the security of the protocol. Many protocols require that some fraction of participants must collude before their security properties can be broken, and it may even be possible to strengthen the security of the protocol by introducing additional participants. This latter approach relies on the observation that it is less probable that a large number of participants (potentially with different legal constraints, economic incentives, concerns about their reputation, and so on) will all choose to collude.

In our experience, many opportunities exist for beneficial and advantageous uses of MPC even without technical safeguards to prevent malicious adversarial behavior [4]. Other incentives and constraints can often reduce the likelihood that parties will collude or deviate from the protocol. For example, parties are likely to be using MPC in order to avoid the liability of holding sensitive data within their own organizations; such parties are unlikely to risk deviating from the protocol because they may then find themselves in possession of sensitive data. As another example, an organization's sole business and reason to exist in the marketplace may be to enable MPC computations [11]; any deviation on the part of such an organization would put its entire business enterprise in jeopardy. Most importantly, these risks can be reduced by requiring participating organizations to commit contractually to adhere to the protocol specification and not to take steps that could compromise the security properties of the protocol.

2.2 Interoperability with Other Privacy-Enhancing Techniques

Many related cryptographic techniques, including both those that have been ubiquitous for decades and those that are as novel as MPC, can be integrated with MPC. We review a few of them below and discuss how they are distinct from MPC but can be complementary in appropriate scenarios.

2.2.1 Symmetric and Public-Key Cryptography

MPC techniques can be combined in a variety of ways with existing workflows that rely on symmetric and public-key cryptography. For example, within an MPC workflow it is possible to encrypt the result of the computation using the public key of the intended recipient *before* secret shares are reconstructed. Alternatively, secret shares themselves can also be encrypted when they are delivered to MPC participants or stored by those participants during the course of the computation. In these scenarios, any third party attempting to access the data involved in the computation might need *both* to gain access to all the secret shares *and* defeat the traditional encryption techniques being employed.

Symmetric and public-key cryptography can also allow secret-sharing schemes to be deployed in scenarios in which some or most participants have limited computing resources [9, 12]. More generally, combining MPC protocols with well-established cryptographic techniques can allow parties to leverage MPC in a broader range of scenarios than would otherwise be possible, and may also introduce opportunities to optimize for performance, communications, number of participants, or other metrics.

2.2.2 Differential Privacy

Differential privacy (DP) techniques make it possible to protect individual records within a dataset while still allowing aggregate analyses over that dataset to be computed and shared [7]. Thus, MPC and DP protect data in complementary but orthogonal ways. An MPC service or workflow protects all input and intermediate data within a computation, ensuring that only the output is revealed to the designated recipients. However, anything the output implies (*e.g.*, about individual records in the original inputs) is also necessarily revealed to the recipients. DP does not by itself protect the inputs or intermediate values during computation. However, if a DP computation is performed in a trusted environment and only the output is shared with a recipient, the recipient's ability to learn about individual records in the original dataset will be limited. It is possible to achieve the benefits of both MPC and DP by performing a DP computation under MPC.

2.2.3 Blockchain

MPC techniques are also complementary to – but independent from – blockchain solutions. A blockchain solution allows individuals and organizations to collectively maintain distributed ledger that is verifiable and cannot be modified [13]. If a scenario requires both (1) the ability to store and compute over secret-shared data and (2) permanent storage and/or verifiability of secret-shared data, it may make sense to combine the two technologies. A number of organizations are working to make the two technologies available in an integrated manner [18]. It is worth noting that MPC techniques and blockchain solutions both come with their own respective forms of overhead in terms of communication, storage, and performance; overheads from the two are cumulative.

3 Software Libraries and Applications

MPC libraries and software solutions developed and offered by Nth Party leverage and combine multiple MPC techniques in order to accommodate a variety of deployment scenarios, including different service architectures and data workflows, different operating environments and software stacks, and different performance constraints. These include libraries in JavaScript, Python, and C++ for variants of MPC techniques described in Section 2.1, facilities and APIs for combining these into larger applications, and complete solutions for particular functionalities such as private set intersection and aggregate data analysis.

Three of the functionalities for which the existing MPC tools have already been deployed in practice or for which the tools are mature and production-ready are (1) aggregate data analysis, (2) private set intersection, and (3) joint computation of a function. Earlier versions of some of these functionalities have been deployed individually as production web applications [14]. These can also be combined to create more sophisticated capabilities, such as support for a small ensemble of relational queries over a dataset [17].

4 Deployment Scenarios, Use Cases, and Regulatory Compliance

Opportunities to deploy an MPC solution can be classified along a number of dimensions, as contemporary MPC technologies (1) can accommodate a number of different types of scenarios in any vertical or industry, (2) can be used today to implement functionalities in a number of categories, (3) can allow organizations to satisfy regulations governing the acquisition, distribution, and use of data in a way that potentially reduces risks and costs, and (4) can be combined with contractual promises not to engage in any external activities that would weaken the security properties of these technologies.

4.1 Configurations and Scenarios

MPC solutions can be useful in a variety of scenarios involving services and workflows that rely on computation over data from multiple contributors. Furthermore, MPC solutions can allow new organizations to contribute their resources and infrastructure to enable or scale up collaboration opportunities while protecting the existing organizations from any additional exposure of their data to the newcomers *and* protecting the newcomers from the liabilities of holding sensitive data [9]. Figure 2 illustrates in an abstract way some of the possible deployment configurations that can allow organizations to leverage and/or enable MPC workflows. The flexibility to accommodate these various configurations is achieved in some cases by using one or more MPC techniques from among those enumerated in Section 2.1.3 in conjunction with established cryptographic techniques in the manner described in Section 2.2.1. Any of these configurations may be applicable to a scenario that falls into one of the categories below.

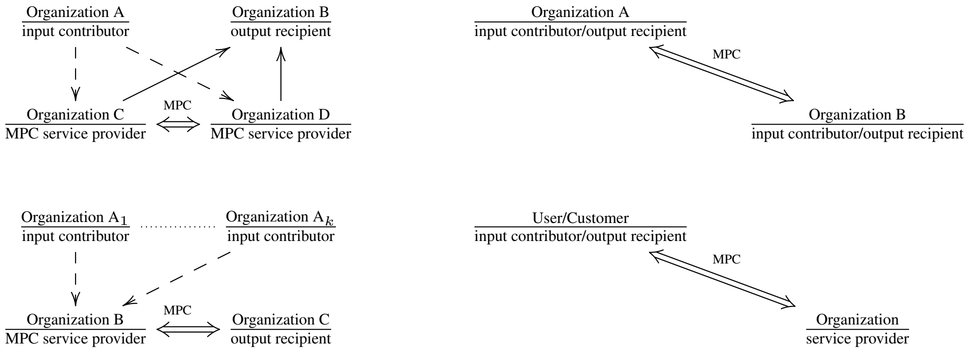

Figure 2: Abstract diagrams of various MPC deployment configurations and the roles of participants within them: (1) two or more organizations may take secret shares of inputs from one party and return secret shares of outputs to another party (top left), (2) two organizations may jointly compute on their combined inputs and then both receive the result (top right), (3) a service provider and output recipient may compute an aggregate over data from many contributors (bottom left), or (4) an organization may compute over user data without seeing the user input and such that only the user can see the result (bottom right).

Internal. Within an organization, multiple units may be subject to constraints or competitive pressures that prohibit them from sharing data. MPC can allow multiple units to nevertheless make decisions, perform benchmarking, and collaboratively compute over their combined datasets while adhering to these constraints.

Business-to-Business. Two or more organizations may deploy MPC techniques to jointly compute over data, or an organization may offer its customer organizations a service that does not require it to share its own data with those customer organizations and does not require that customer organizations share their data with the service provider.

Business-to-Customer. Businesses may offer services over the web that store and operate on customer data without disclosing that data to the service provider. This may be achieved by allowing the web browsing application being used by the customer on their own device to act as the second party in an MPC workflow, or by having the service provider leverage a third organization (or any additional number of organizations) to make the service secure via MPC. Note that due to the properties of MPC, the additional organizations do not see any of the data, either.

4.2 Use Cases

While there exist many opportunities to use MPC within existing services and workflows, there are a few specific opportunities worth identifying that are well-suited *today* for existing MPC solutions in terms of usability, scalability, and maturity of MPC techniques.

Safe and Cost-Efficient Decision-Making. MPC can help decision-makers analyze data to answer simple questions (*e.g.*, with “yes” or “no” answers) without first undertaking a burdensome negotiation process that may involve legal expenses, delays, and risks of data exposure or unauthorized data reuse.

Alternatives to Clean Rooms. For computations over data sources that have consistent formats and schemas, MPC solutions can provide a less expensive alternative to clean rooms and trusted third parties. Two (or more) organizations can each procure an MPC tool (packaged as a virtual machine, container, or software package), set up their respective tool instances within their own environments, and allow these tool instances to communicate with one another via MPC. Alternatively, an organization such as Nth Party can provide SaaS MPC solutions without seeing any of the data being analyzed by the users of the service.

Aggregate Analysis, Evaluation, and Benchmarking. With MPC, multiple organizations can leverage aggregate data analysis workflows on pooled data without sharing any of the constituent datasets. Industry competitors can run fully confidential surveys across customers or others to create aggregate benchmarks; this can add value similar to that of a "credit bureau" for customer behaviors or can be used to validate industry research. Partner organizations can also evaluate the value or effectiveness of their integrations and partnerships, such as calculating conversion rates or identifying response correlations across audience segments and outreach strategies.

Privacy-Preserving Services. Existing customer-facing web services offered by organizations can be enhanced to be privacy-preserving, so that customers can still enjoy the value of the service while not sharing their data with the service provider. On the other side, service providers can reduce or eliminate the liability (and possibly regulatory compliance costs) associated with storing, making available securely, and destroying customer data.

4.3 Compliance with Data Protection and Privacy Regulations

MPC solutions broaden the spectrum of options available to organizations when operating in regions or handling datasets that are subject to data protection and privacy regulations. The General Data Protection Regulation (GDPR) within the European Union [6] and the California Consumer Privacy Act (CCPA) in the US state of California [2] impose requirements on organizations that collect or process personal data of individuals. Use of MPC for consumer-facing online services can be leveraged to avoid triggering compliance obligations under these laws or to reduce compliance obligations, and can extend the utility of data that is collected and stored in accordance with such requirements.

Pseudonymization. The GDPR suggests that entities that collect personal data may use pseudonymization techniques to help comply with their security obligations under that law. Article 4(5) defines pseudonymization as "the processing of personal data in such a way that the data can no longer be attributed to a specific data subject without the use of additional information" that must be kept separate from the pseudonymized data [6]. MPC can be used to achieve pseudonymization when it is used to transform data into secret shares across two units or organizations, provided that the parties sharing the information do not collaborate. Because MPC also allows the client-side application or software (*e.g.*, a mobile application or webpage running on a user's personal device) to maintain secret shares of data submitted by the user, MPC can be used to ensure that additional information that could identify a user is never available in any form to the organization providing the online service. Furthermore, unlike pseudonymization techniques such as tokenization, MPC places no restrictions on the complexity of a computation on pseudonymized data and also creates no persistent pseudonymized data during the course of that computation that may remain in the possession of the parties involved. Thus, organizations can offer a greater variety of services to customers while maintaining compliance.

De-Identification, Anonymization, and Data Processing. In some services and data workflows between an organization and another partner organization, the output of the process may satisfy the definition of *de-identified* data (at the level of an individual record) or *anonymized* data (at the aggregate level), which is not subject to compliance obligations under laws governing the protection of personal data such as the GDPR [6] and CCPA [2]. However, in order to produce this de-identified or anonymized output, the service or workflow may require record-level access to personal data being controlled by the organization, and it might then need to compute over this granular data together with propriety, private, or sensitive data from the partner organization. In the absence of MPC, performing the calculations necessary to obtain the de-identified or anonymized output might require at least (1) that both organizations incur the compliance costs and liability associated with transferring personal data to the partner organization or (2) that the partner organization share its proprietary, private, or sensitive data to enable the computation. By instead using MPC to perform the calculations necessary to obtain and deliver the aggregate anonymized data or the record-level de-identified data to the intended recipient, the respective input datasets of the two organizations can remain solely within their own possession throughout. Accordingly, organizations can use MPC, in conjunction with contractual commitments between the partner organizations to not "re-identify" or share data, to share only de-identified or anonymized data and thus reduce compliance burdens.

Erasure and Opt-out. Collecting, storing, and processing personal data or cryptographic keys in a secret-shared form (where the secret shares are distributed either between separate organizations or between the organization and the individual from whom the data was collected) can be beneficial in a variety of ways to an organization. As one example, we consider an organization that always stores personal data in encrypted form across its infrastructure and maintains the decryption keys in secret-shared form between itself and another organization. Because the organization cannot on its own decrypt any data in its possession, the partner organization can at any time fulfill an individual's data erasure request simply by destroying their secret shares for the decryption keys for that individual. As another example, we consider an organization that allows users to make data subject access requests or to opt out of some service by filling out a form on a website. By using MPC to run the process of determining whether the organization possesses an individual's data in the first place (*e.g.,* by performing a database lookup using the personal data supplied by the individual while keeping it in encrypted or secret-shared form), the organization can offer this service without incurring any additional burdens associated with collecting the individual's data.

References

[1] Accessible and Scalable Secure Multi-Party Computation. `https://multiparty.org/`.

[2] The California Consumer Privacy Act of 2018. `https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180AB375`.

[3] M. Bellare, V. T. Hoang, and P. Rogaway. Foundations of garbled circuits. Cryptology ePrint Archive, Report 2012/265, 2012. `https://eprint.iacr.org/2012/265`.

[4] A. Bestavros, A. Lapets, and M. Varia. User-centric distributed solutions for privacy-preserving analytics. *Communications of the ACM*, 60(2):37–39, February 2017.

[5] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin, and M. Y. Zhu. Tools for privacy preserving distributed data mining. *SIGKDD Explor. Newsl.*, 4(2):28–34, Dec. 2002.

[6] Council of European Union. Council regulation (EU) no 2016/679, 2016. `https://eur-lex.europa.eu/legal-content/EN/TXT/?qid=1589662968663&uri=CELEX:32016R0679`.

[7] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, Aug. 2014.

[8] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.

[9] F. Jansen, K. D. Albab, A. Lapets, and M. Varia. Accessible Privacy-Preserving Web-Based Data Analysis for Assessing and Addressing Economic Inequalities. In *Proceedings of ACM COMPASS 2018: First Conference on Computing and Sustainable Societies*, San Jose, CA, USA, June 2018.

[10] S. Jarecki and X. Liu. Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection. In *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*, pages 577–594. Springer, 2009.

[11] A. Lapets, K. D. Albab, R. Issa, L. Qin, M. Varia, A. Bestavros, and F. Jansen. Role-Based Ecosystem for the Design, Development, and Deployment of Secure Multi-Party Data Analytics Applications. In *2019 IEEE Cybersecurity Development (SecDev)*, McLean, VA, USA, September 2019.

[12] A. Lapets, N. Volgushev, A. Bestavros, F. Jansen, and M. Varia. Secure MPC for Analytics as a Web Application. In *2016 IEEE Cybersecurity Development (SecDev)*, pages 73–74, Boston, MA, USA, November 2016.

[13] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, USA, 2016.

[14] L. Qin, P. Flockhart, A. Lapets, K. D. Albab, M. Varia, S. Roberts, and I. Globus-Harris. From usability to secure computing and back again. In *Proceedings of the 15th Symposium on Usable Privacy and Security (SOUPS)*, Santa Clara, CA, USA, August 2019.

[15] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

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

[17] N. Volgushev, M. Schwarzkopf, B. Getchell, M. Varia, A. Lapets, and A. Bestavros. Conclave: Secure Multi-Party Computation on Big Data. In *Proceedings of EuroSys 2019: The 12th European Conference on Computer Systems*, Dresden, Germany, March 2019.

[18] G. Zyskind, O. Nathan, and A. Pentland. Enigma: Decentralized computation platform with guaranteed privacy. *CoRR*, abs/1506.03471, 2015.