UNIQUE SOFTWARE ENGINEERING TECHNIQUES: PANACEA FOR THREAT COMPLEXITIES IN SECURE MULTIPARTY COMPUTATION (MPC) WITH BIG DATA

Uchechukwu Emejeamara[1], Udochukwu Nwoduh[2] and Andrew Madu[2]

[1]IEEE Computer Society, Connecticut Section, USA
[2]Department of Computer Science, Federal Polytechnic Nekede, Nigeria.

ABSTRACT

Most large corporations with big data have adopted more privacy measures in handling their sensitive/private data and as a result, employing the use of analytic tools to run across multiple sources has become ineffective. Joint computation across multiple parties is allowed through the use of secure multi-party computations (MPC). The practicality of MPC is impaired when dealing with large datasets as more of its algorithms are poorly scaled with data sizes. Despite its limitations, MPC continues to attract increasing attention from industry players who have viewed it as a better approach to exploiting big data. Secure MPC is however, faced with complexities that most times overwhelm its handlers, so the need for special software engineering techniques for resolving these threat complexities. This research presents cryptographic data security measures, garbed circuits protocol, optimizing circuits, and protocol execution techniques as some of the special techniques for resolving threat complexities associated with MPC's. Honest majority, asymmetric trust, covert security, and trading off leakage are some of the experimental outcomes of implementing these special techniques. This paper also reveals that an essential approach in developing suitable mitigation strategies is having knowledge of the adversary type.

KEYWORDS

Cryptographic Data Security, Garbed Circuits, Optimizing Circuits, Protocol Execution, Honest Majority, Asymmetric Trust, Covert Security, Trading Off Leakage.

1. INTRODUCTION

In recent years, the issue of data security has continued to attract global attention. More people are becoming informed of the significance of protecting their privacy. In fact, the cases of Google and Facebook which have been under scrutiny illustrate the trend towards enhancing confidentiality and privacy in society. Software engineering techniques are multiple ways of approaching software development and delivery [8]. A threat, in software engineering, is an application or malicious code that can cause damage to the computer or steal personal data [2]. The need for security measures to safeguard data has been necessitated by the increase in the use of technology in the public and private sectors [7]. Moreover, concerns have been raised by both private and public sectors on the data mined by data mining tools. The application of these data mining tools has conflicted with the privacy policies of individuals.

David C. Wyld et al. (Eds): NLP, JSE, MLTEC, DMS, NeTIOT, ITCS, SIP, CST, ARIA - 2020
pp. 169-176, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101413

The garbed circuit is one of the most secure multi-party computation techniques that can be used in securing each party's contribution in instances when two or more parties need to compute a given common result. Trusted execution environments offer data and code-based hardware-implemented seclusion. The seclusion process makes these environments trusted candidates and this makes the secure multi-party tractable. These users can execute their contributions privately and only reveal their output. Overall, unique software engineering techniques offer solutions for the threat complexities in secure multi-party protocol computation with big data.

Private inputs from different parties that do not trust each other have facilitated the development of multi-party computation. Threat complexities propagate due to less on no knowledge concerning the multi-party computations. This paper also has addressed approaches used to mitigate these issues, including adaptive adversaries and static exploration. In multiparty protocol computation, recovering the reuse of a once-corrupted component can act as a solution to adaptive adversaries. Thus, despite some implications, the exploitation of big data justifies the implementation of multi-party protocol computation.

2. APPROACHES TO SECURE COMPUTATION

The three main approaches used in secure computation include homomorphic encryption, secret sharing, and Yao's garbled circuit [17].

2.1. MPC Based On Secret Sharing

Secret sharing enables the computation of information privately. In this case, a secret scheme is relied upon by the involved parties in carrying out the computations [17]. This means that no information relating to the data is revealed as the private data is provided in random values. The advantage of this technique is that it does not require any encryption key and allows information-theoretical security [18]. However, it requires continuous communication among the involved parties.

2.2. MPC Based On Homomorphic Encryption

Homomorphic encryption provides an approach to manipulate data while maintaining confidentiality. Using this approach, parties rely on a homomorphic encryption scheme like the Paillier scheme in the encryption of data [17]. Using the encrypted data, the parties are able to perform computations. Over the last years, the technique has attracted attention owing to its ability to preserve privacy. The technique is pegged on the complexity of the problem at hand [15]. While this feature can be viewed as strength, it can also be regarded as a drawback as it leads to difficulty in dealing with complex problems.

2.3. MPC Based On Yao's Garbled Circuits

Yao's garbled circuit a form of a function that allows communication between two or more parties without infringing on their privacy [16]. When using this approach, one party can encrypt data (input) and carry out computations. The resultant output (computed input) is converted into a circuit and presented in the form of a binary gate. The encrypted input is then sent to the other party in form of a circuit and by evaluating this input; the other party is able to decipher it to generate output – by comparing the bits and combining the results [15]. The approach is regarded as the most efficient since it does not call for continuous communication between the parties involved.

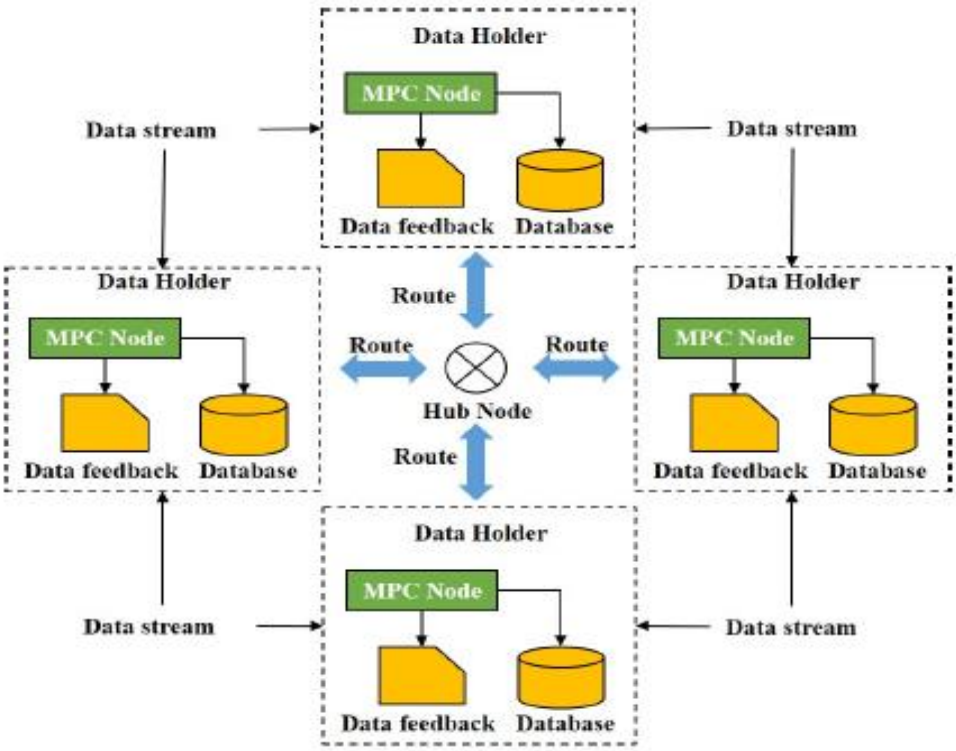

Figure 1. MPC technical framework [16]

Figure 1 above shows the MPC technical framework. The figure illustrates the transmission of data among parties and the specific areas that need to be secured to ascertain the privacy and confidentiality of information. This is where the three approaches of MPC play a key role. A hub node transmits the network and signals control when the MPC computing task is launched. Each of the data holders can commence a collaborative computing task. For secure collaborative computing, addresses are routed through hub nodes and the remaining data holders (of similar data types) are selected for computation. MPC nodes of multiple data holders participate in collaborative computing, query the required data from the databases based on the calculation logic. On the basis of ensuring privacy, all the involved parties get the correct feedback and no data is leaked to any of the participants [16].

3. SECURE MULTIPARTY COMPUTATION (MPC) WITH BIG DATA

In modern business processes, big data analysis has become a significant development that has helped in generating accurate results and accessing private data from different sources. A lot of companies have adopted more privacy measures on their own private data and as a result, employing the use of analytic tools to run across multiple sources has become ineffective. Big data is a field that deals with ways of analyzing complex or too large sets that cannot be handled by traditional data-processing techniques [10]. Joint computation across multiple parties is allowed through the use of secure multi-party computations. This understanding enables the communication of parties without them having to disclose or reveal their private data input.

Implementation of the multi-party protocol computations are only done on larger workflows [9]. There are several challenges that the implementation of the multi-party protocol computation faces. These challenges include poor integration of data processing systems with analytics and

multi-party computation [1], requiring significant expert knowledge to be able to run analytics in the multi-party computation framework [6], the incapability of the multi-party computation to support data-parallel process outside the multi-party computations, and poor scaling of frameworks to large data sets. Therefore, the viability of multi-party computation implementation can be established through addressing the challenges, application, and adversaries of the multi-party computations.

Different data owners can be united through secure multi-party in function computation that depends on their data even if they might be having trust issues with each other and this is the primary goal of secure multi-party computation [3]. In multi-party computation, all the participants involved are data input owners. The essential roles of the multi-party computation include the IP (which belongs to the input parties), and they are responsible for sending data to the private computation, and result parties (RP) who receives the result from private computation and the computation parties (CP) whose responsibility is to carry out joint private computations [11]. The most common protocol of the multi-party computation is the lack of a single trust point and it involves many organizations and persons. Due to this, access to encrypted data is prohibited to all the computing parties, and no party can access the data.

There are many cases where MPC can prove worthwhile. However, for the purpose of this paper, two examples are used to show the application of MPC on big data.

3.1. Credit Card Regulation

The financial industry is one of the biggest beneficiaries of big data. Collaboration between the regulators and the industry players can enhance the efficiency of the sector by relying on data sharing and analysis. A government regulator overseeing the consumer credit reporting may wish to estimate the credit score of the consumers based on their geographic region. In this case, the government has social security numbers and the ZIP codes for all the citizens. However, the credit reference organizations have the SSNs of the credit cardholders, credit lines, and credit ratings. As required by law, the government cannot share private information with other parties [17], [15]. Similarly, credit organizations cannot share customer data with external parties. Thus, in this case, MPC is needed to facilitate data analysis and decision making.

3.2. Market Concentration

The law requires the government to regulate the market and avoid monopolies or oligopolies. In most cases, regulators use the Herfindahl-Hirschman Index (HHI) – which is based on the sum of the squared market shares of organizations in the active market. Based on the analysis, a government decides whether scrutiny is required. While public revenue data is easy to obtain, privately-held information is not easily accessed. In this regard, to effectively carry out the analysis, it is paramount to use MPC owing to the presence of private data in various agencies [17], [15]. Moreover, MPC makes it possible to filter and aggregate millions of records that organizations keep confidential.

3.3. Security Guarantees

The use of MPC ascertains the privacy of computed input and intermediate data. The way it works is that MPC does not reveal what is flagged as sensitive data. Moreover, it tends to ensure that the correctness of the output attained is within the standards accepted by all the parties [15].
The output and input processing need to adhere to the set speeds and credibility standards. However, it is required that all the participants must adhere to the regulations – they must be honest in their dealings to ensure that the privacy and confidentiality of the shared data are

maintained at all times. The generally accepted standard is that all the parties must exhibit uttermost honesty in their engagement [15].

4. MANAGING THREAT COMPLEXITIES IN SECURE MULTIPARTY COMPUTATION (MPC) WITH BIG DATA

To enhance security, there are several properties that the multi-party computation employs, and they also help to enhance the efficiency and robustness of the system [4]. In the multi-party system, the most common protocol is that there is no single point of trust. The other most important protocols are n, f, passive security, abort active security and fault tolerance active security. n represents the number of computing parties involved, *f represents* the maximum number of computing parties allowed to regulate and run the protocol intended in which f+1 will be a violation of the system, passive security provides a guarantee to the privacy of source data such as the number of computation parties involved, *abort active security* ensures that corrupt computational parties run the purported protocol and *faulty tolerance active security* has the role of ensuring continuous operation of the system even in instances when the computational parties have ceased to operate correctly [5]. Figure 2, shows the structure of a secure MPC and how the various parties involved compute a function using their inputs, while keeping these inputs private [14].

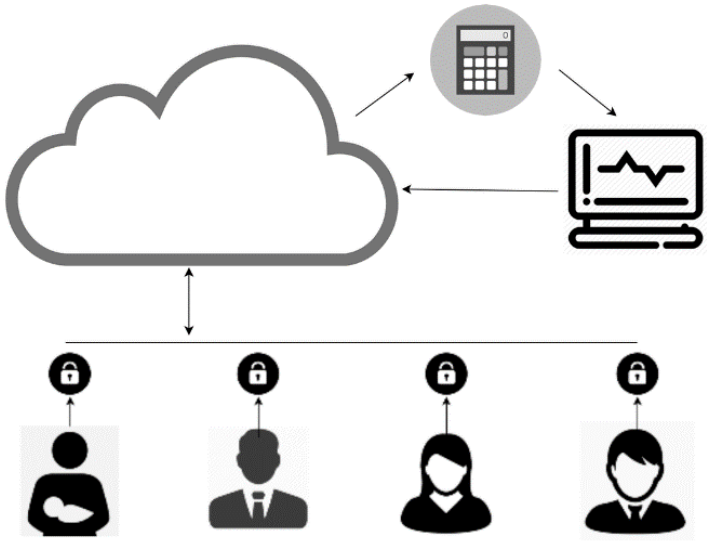

Figure 2. Secure Multiparty Computation [14].

Many trusted entities can be used to create a neural multi-party computation since it does not depend on the organization or individual's trustworthiness. The other solution that can result in the creation of a more reliable multi-party computation is the combination of entries with conflicting interests and trustworthiness entities. In multi-party computation, the security of the system against threats is a vital consideration. A security attack on multi-party computation can result in many adversaries being attacked since the system deals with big and private data computations. The adversaries that can be attacked can be classified into malicious, covert, and semi-honest since they involve encrypted inputs. Computational security and statistical security are other mitigation techniques that can be used to avoid multi-party computation complexities.

4.1. Cryptographic Data Security

Most large corporations have employed the use of traditional methods of cryptography in their data security. In large corporations, a lot of cryptographic tasks that are diverse in nature and a given number of keys are assigned to individuals to manage them. In terms of control, storage, and usage of these keys, it is carried out under different circumstances in what is referred to as hardware security modules [12]. Companies employ the use of dedicated hardware security modules that provide cryptographic operations across the entire corporation. Exportation or incorporation into the hardware security is the most operations that are done to cryptographic keys. Techniques referred to as key-wrap are used to develop these keys. After the keys have been generated by the hardware security module disintegration to the design is not possible. To maintain the security of the keys, a lock is put on them by the hardware security module and they are safeguarded by key-wrap technique. A call to several cryptographic operations can be made due to the availability of embedded keys on the hardware security module. The hardware security module also backs up the standard cryptographic API. Figure 3 is a cryptographic data security protection platform that features multiple data security products that can be deployed individually or in combination to deliver advanced encryption, tokenization, and centralized key management [13].

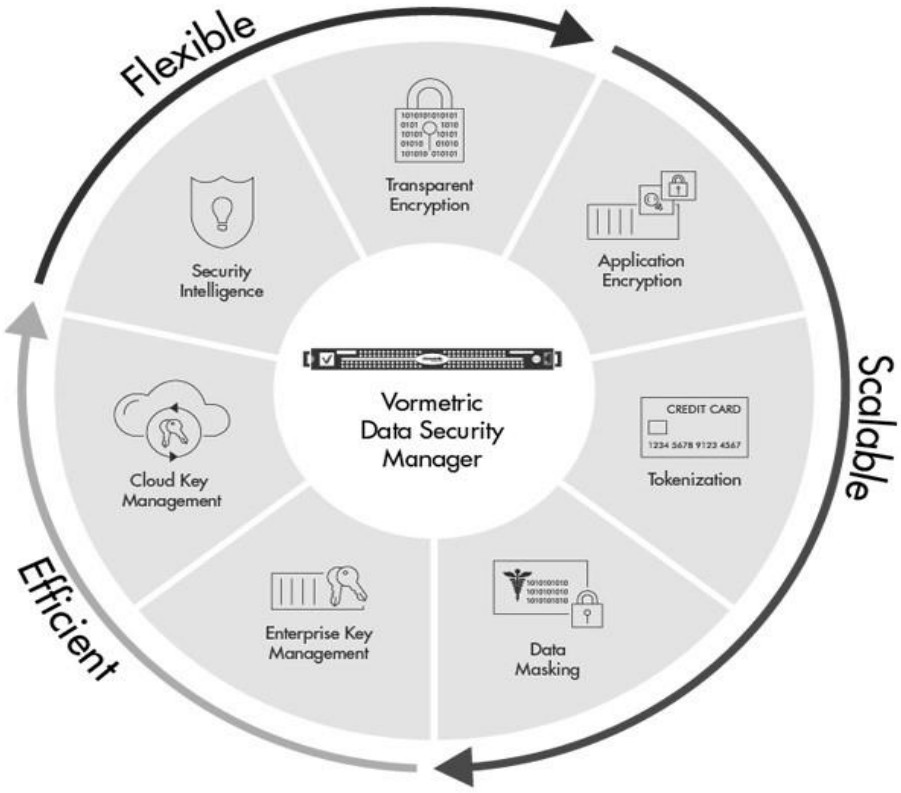

Figure 3. Vormetric Data Security Platform [13].

4.2. Implementing Other Special Software Engineering Technique

Some software engineering techniques are viably utilized in resolving threat complexities in multi-party communications. One of the techniques that can be used is less expensive garbling. Garbed circuits protocol is a special technique used, in which the main cost of executing the circuits is on the computation that is required to evaluate and generate garbed tables and the

bandwidth that is required in the transmission of the garbed gates. Traditional garbling methods have been improved in terms of the bandwidth and the computation to evaluate and generate the garbled gates.

Another technique is optimizing circuits which involves reducing the size of the circuit to make a direct impact on the protocol cost. Protocol execution is another implementation technique used which addresses various improvements on the way multi-party communication protocols are executed. In this technique, the scaling issues are addressed and eliminated. Lastly, many programming tools are used in the implementation of these techniques in handling threat complexities in multi-party computations. These tools vary in different ways such as their input languages, the protocol they support, and how they combine input programs into circuits.

4.3. Research Results

Several outcomes exist from the implementation of software techniques in resolving threat complexities multi-party computation. Honest majority is one of the experimental results in which the adversary is likely to corrupt less than two of all parties involved since functions in the system have information-theoretically secure protocols. Asymmetric trust is another experimental outcome that may be central to the standard assumption of all parties being equally distrusting. The other experimental result is covert security which is reasonable in many settings but may be insufficient for some applications. A party can deviate from the protocol and be caught in a fixed probability which is referred to as public verifiable covert. Finally, trading off leakage is seen as an experimental result from the implementation of the multi-party computation. The use of these special software engineering techniques in resolving threat complexities in secure multi-party computation provides very strong security guarantees at a given cost and these security computations may make it difficult for hackers to gain access to a single bit of data.

5. CONCLUSION

In this technology-driven world, big data computations, and business analytics have attracted many adversaries, which tends to access business entities and private data. Due to this trend, developing and implementing special software engineering techniques are very essential in resolving threat complexities encountered in multi-party computation with big data. To ensure high-security levels of multi-party computations, regulations and protocols should be implemented properly. The other essential approach in developing suitable mitigation strategies is having knowledge of the adversary type. This research has clearly shown that some expertise and experience are required to adhere to specific protocols, and by implementing all these techniques, resolving threat complexities in multi-party computations with big data will be feasible and super-efficient.

REFERENCES

[1] Alam, K. S., Xiao, D., Akter, M. P., Zhang, D., Fletcher, J., & Rahman, M. F. (2018). Modified MPC with extended VVs for grid-connected rectifier. *IET Power Electronics*, *11*(12), 1926-1936.

[2] Ansari, M. T. J., Pandey, D., & Alenezi, M. (2018). Store: Security threat-oriented requirements engineering methodology. *Journal of King Saud University-Computer and Information Sciences*.

[3] Archer, D. W., Bogdanov, D., Lindell, Y., Kamm, L., Nielsen, K., Pagter, J. I., ... & Wright, R. N. (2018). From keys to databases—real-world applications of secure multi-party computation. *The Computer Journal*, *61*(12), 1749-1771.

[4] Evans, D., Kolesnikov, V., & Rosulek, M. (2017). A pragmatic introduction to secure multi-party computation. *Foundations and Trends® in Privacy and Security*, *2*(2-3).

[5] Hastings, M., Hemenway, B., Noble, D., & Zdancewic, S. (2019, May). Sok: General purpose compilers for secure multi-party computation. In *2019 IEEE Symposium on Security and Privacy (SP)* (pp. 1220-1237). IEEE.

[6] Houska, B., & Villanueva, M. E. (2019). Robust optimization for MPC. In *Handbook of Model Predictive Control* (pp. 413-443). Birkhäuser, Cham.

[7] Lykou, G., Anagnostopoulou, A., & Gritzalis, D. (2018, June). Implementing cyber-security measures in airports to improve cyber-resilience. In *2018 Global Internet of Things Summit (GIoTS)* (pp. 1-6). IEEE.

[8] Mezhuyev, V., Al-Emran, M., Ismail, M. A., Benedicenti, L., & Chandran, D. A. (2019). The acceptance of search-based software engineering techniques: An empirical evaluation using the technology acceptance model. *IEEE Access*, *7*, 101073-101085.

[9] Montazeri-Gh, M., Rasti, A., Jafari, A., & Ehteshami, M. (2019). Design and implementation of MPC for turbofan engine control system. *Aerospace Science and Technology*, *92*, 99-113.

[10] Oussous, A., Benjelloun, F. Z., Lahcen, A. A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University-Computer and Information Sciences*, *30*(4), 431-448.

[11] Tso, R., Alelaiwi, A., Rahman, S. M. M., Wu, M. E., & Hossain, M. S. (2017). Privacy-preserving data communication through secure multi-party computation in healthcare sensor cloud. *Journal of Signal Processing Systems*, *89*(1), 51-59.

[12] Yellepeddy, K. K., Peck, J. T., Hazlewood, K. M., & Morganti, J. A. (2017). *U.S. Patent No. 9,794,063*. Washington, DC: U.S. Patent and Trademark Office.

[13] "Vormetric Data Security Platform," *Thales*. [Online]. Available: https://cpl.thalesgroup.com/encryption/vormetric-data-security-platform. [Accessed: 03-Jul-2020].

[14] Originally published by Shaan Ray on, "What is Secure Multi Party Computation?," 09-Jun-2020. [Online]. Available: https://hackernoon.com/what-is-secure-multi-party-computation-232caef900b9.

[15] Volgushev N., Schwarzkopf M. and Getchell B., "Conclave: secure multi-party computation on big data: Extended Technical Report", Conference Paper, 2018. [Accessed 28 January 2020].

[16] Yan S., Liu C., Wang M., Ma P., and Wei K., "Promoting Data Circulation by Secure Multi-party Computation and Blockchain", DEStech Transactions on Computer Science and Engineering, no., 2018. Available: 10.12783/dtcse/ceic2018/24542.

[17] Raeini M. G. and Nojoumian M., "Privacy-Preserving Big Data Analytics: From Theory to Practice", Security, Privacy, and Anonymity in Computation, Communication, and Storage, pp. 45-59, 2019. Available: 10.1007/978-3-030-24900-7_4 [Accessed 28 January 2020].

[18] Ankele R., Kucuk K., Martin A., Simpson A., and Paverd A., "Applying the Trustworthy Remote Entity to Privacy-Preserving Multiparty Computation: Requirements and Criteria for Large-Scale Applications", 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016. Available: 10.1109/uic-atc-scalcom-cbdcom-iop-smartworld.2016.0077 [Accessed 28 January 2020].