Collective Threshold Multiparty Private Set Intersection Protocols for Cyber Threat Intelligence

Guan, C.; van Assen, J. S.; Erkin, Z.

DOI
10.1109/WIFS61860.2024.10810671

Publication date
2024

Document Version
Final published version

Published in
Proceedings - 16th IEEE International Workshop on Information Forensics and Security, WIFS 2024

Collective Threshold Multiparty Private Set Intersection Protocols for Cyber Threat Intelligence

C. Guan, J.S. van Assen and Z. Erkin
Cyber Security Group
Delft University of Technology
Van Mourik Broekmanweg 5, 2828 XE, Delft, The Netherlands
c.guan-1@student.tudelft.nl, j.s.vanassen@tudelft.nl, z.erkin@tudelft.nl

2024 IEEE International Workshop on Information Forensics and Security (WIFS) | 979-8-3503-6442-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/WIFS61860.2024.10810671

***Abstract*—Multiparty private set intersection enables multiple parties to determine the intersection of their private sets without disclosing the actual content. It is pivotal for collaboration in cyber threat intelligence as it allows organizations to share compromising or sensitive data in a privacy-preserving manner. This data includes infected IP addresses, malware hashes and other indicators of compromise. Then, the organizations identify elements that overlap across all datasets and take action to mitigate the threat with the broadest impact. Although, in many cases, the condition that an element be present in all sets is too stringent. Therefore, in this work, we focus on *threshold multiparty private set intersection* (T-MPSI), a protocol that identifies elements present in a subgroup of the total sets instead of in all sets. We highlight the differences between three different perspectives when computing the threshold intersection: individual—only the party leader learns the elements from their set that meet the threshold, all—all parties learn the elements from their set that meet the threshold, and collective—all parties jointly learn all elements that are present in the threshold, regardless of whether they possess those elements themselves.**

While many implementations for T-MPSI$_{\text{individual}}$ and T-MPSI$_{\text{all}}$ have been proposed, to the best of our knowledge, no implementation for T-MPSI$_{\text{collective}}$ exists. Therefore, we present a generic composition that extends any T-MPSI$_{\text{individual}}$ protocol into a T-MPSI$_{\text{collective}}$ protocol. Our extension employs a multiparty private set union to aggregate outputs efficiently. We then provide a comprehensive analysis and runtime evaluation, demonstrating the feasibility of the extension.

I. Introduction

Multiparty private set intersection (MPSI) protocols allow multiple parties to determine the intersection of their private sets without disclosing the sets themselves, ensuring that each party's data remains confidential throughout the process [1]. These protocols have a wide range of potential practical applications [2], [3], [4]. For instance, they are utilized in targeted advertising to match customer preferences without revealing personal data. In social and messaging platforms, these protocols help identify mutual contacts in a secure manner. MPSI is also used to coordinate available times across private calendars to facilitate scheduling meetings.

In cyber security, MPSI protocols are essential for sharing incident information related to *cyber threat intelligence* (CTI). It plays a crucial role in keeping up with evolving cyber threats, including ransomware attacks [5] and phishing campaigns [6]. CTI involves the collection, analysis, and dissemination of information about potential or current cyber threats and vulnerabilities. It provides experts with the necessary insights to enhance situational awareness for proactive defence and swift incident response [7]. It categorizes threat data into three main types: tactical, operational and strategic [8]. Tactical intelligence deals with indicators of compromise such as infected IP addresses, malware hashes and email headers indicative of phishing scams. Operational intelligence focuses on the *tactics, techniques, and procedures* (TTPs) of cyber criminals, i.e., the exploit paths employed. Strategic intelligence provides a high-level overview of cyber threat trends and motives.

Since threat actors often employ similar attack techniques at different sites, it is useful for entities to share CTI information. MPSI can be employed in CTI sharing in the following example. Organizations maintain lists of suspected malicious IP addresses. By pooling and comparing these suspicious IPs, they can more effectively pinpoint the shared threat. Once identified, these organizations collaboratively block the malicious IP, strengthening their defences against the related malware. Another example is when various organizations decide to exchange TTPs and find a common vulnerability within their software. From this insight, they can prioritize fixing this vulnerability to protect against attacks that exploit this particular weakness.

However, the requirement that an element be present in all parties' datasets in order to be outputted in the intersection can be overly restrictive for many practical applications, including CTI sharing. Consider the scenario where five organizations exchange CTI to identify which suspected IP addresses are most likely malicious. If the same IP address is consistently found across all their datasets, it strongly indicates a potential threat. Moreover, even if an IP does not appear in all datasets but is present in most, say in four out of five, it merits further investigation. In such cases, the parties would benefit from computing the intersections of subgroups of four sets instead of the absolute intersection. This variant of MPSI that outputs the elements which appear in at least a threshold number of datasets T is known as *threshold multiparty private set intersection* (T-MPSI). In some papers, it is referred to as over-threshold MPSI [9], d-and-over MPSI [10] or quorum PSI [11].

Note that T-MPSI protocols are defined variably across

studies, adding complexity to the field. Some definitions of the threshold involve the minimal cardinality of the intersection [12], [13] or the maximal difference between private set sizes [14].

Within our focused definition of T-MPSI, where the threshold refers to the number of sets containing the item, we encounter several operational perspectives.

- **T-MPSI$_{\text{individual}}$**: The party leader learns which of their own elements meet the threshold.
- **T-MPSI$_{\text{all}}$**: All parties learn which of their own elements meet the threshold.
- **T-MPSI$_{\text{collective}}$**: Every parties learns which elements meet the threshold across all sets, not necessarily just their own.

T-MPSI$_{\text{individual}}$ or T-MPSI$_{\text{all}}$ is useful in situations where organizations do not need to learn about any element that is not part of their set, e.g., to identify a common software vulnerability for patching. In this case, each party is only concerned with identifying which of their own software's vulnerabilities are common to the other sets. They would not benefit from learning the T-MPSI$_{\text{collective}}$ intersection since it can include vulnerabilities that do not affect their systems. Another reason T-MPSI$_{\text{collective}}$ is not suitable for this scenario is that an organization may not want their vulnerabilities revealed to others, unless they also share that vulnerability, since it could compromise their security posture. T-MPSI$_{\text{collective}}$ is better suited for preventive measures in CTI. If the sets are defined to contain well-defined hashes or identifiers of CTI data, a participant would want to employ a T-MPSI$_{\text{collective}}$ protocol over a T-MPSI$_{\text{all}}$ protocol to verify the presence of new threats within the collective set intersection. Furthermore, by analyzing shared elements across various security logs, threats that impact multiple organizations—and therefore represent a higher risk—can be identified to uncover common patterns of attacks. Even though an element may not appear in an organization's dataset, they can still choose to take action against it preemptively.

Despite the theoretical advancements within the T-MPSI domain, there seems to be a notable gap in practical implementations, especially for the T-MPSI$_{\text{collective}}$ perspective. Therefore, we show a generic composition that can be applied to any T-MPSI$_{\text{individual}}$ type protocol to turn it into a T-MPSI$_{\text{collective}}$ type protocol by applying a union operation. As of now, T-MPSI$_{\text{individual}}$ and T-MPSI$_{\text{collective}}$ protocols are not directly comparable in terms of computational and communication complexities. Unifying both variants of T-MPSI would address the issue with the lack of T-MPSI$_{\text{collective}}$ protocols since it can broaden the operational scope of any and all T-MPSI$_{\text{individual}}$ protocols.

II. Preliminaries

We introduce the necessary background and definitions to understand the perspectives of T-MPSI and the extension.

A. T-MPSI Perspectives

In current literature, no consensus exists on the exact meaning of T-MPSI. For this work, we use the following definition: T-MPSI is an MPSI protocol which returns items present in a minimum number of the total m sets as specified by the threshold, T [3], [9], [15]. Furthermore, we formally define the different perspectives of T-MPSI as follows.

- **T-MPSI$_{\text{individual}}$**: A party leader j wishes to obtain the elements of their private set i that are also present in at least a threshold $T-1$ of the other $m-1$ parties' sets [3], [10], [11], [15].
- **T-MPSI$_{\text{all}}$**: All parties obtain the elements that are in their private set and at least $T-1$ other sets [9]. It is equivalent to computing the T-MPSI$_{\text{individual}}$ for every party i.
- **T-MPSI$_{\text{collective}}$**: Every party learns which elements appear at least a threshold number of times T in the combined private input of the parties [15].

We will illustrate the nuances between these perspectives with an example. Figure 1 represents three intersecting sets S_1, S_2 and S_3. An absolute intersection in standard set theory would be depicted by the orange section in the centre of the image. However, a threshold intersection differs by including elements that are common to a subgroup of the sets based on a specified threshold. Figure 2 illustrates the various threshold intersections based on the different perspectives. Figure 2 reads as "at the end of a ___ protocol (columns), each party i (rows) learns ___." After running the T-MPSI$_{\text{individual}}$ protocol with party 1 playing the role of the leader, party 1 learns all the elements in their set that are also present in at least $T-1$ of the other sets. Parties 2 and 3 do not learn anything. After running the T-MPSI$_{\text{all}}$ protocol, all parties have taken turns being the leader. Therefore, all parties learn which of the elements in their set are also present in at least $T-1$ of the other sets. After running the T-MPSI$_{\text{collective}}$ protocol, all parties find out which elements—not only the ones in their set—are present in at least T of the sets.

B. Multiparty Private Set Union

In *multiparty private set union* (MPSU), each party holds a private set. The objective is to find the union of these sets without revealing any additional information about the individual sets beyond what can be inferred from the final result, i.e.,

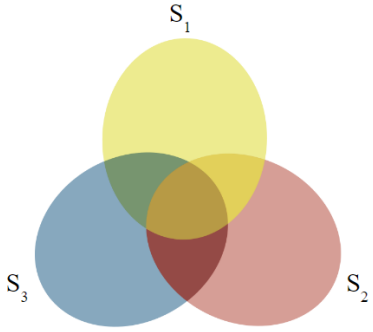

Fig. 1. Three sets S_1, S_2 and S_3 in a T-MPSI protocol.

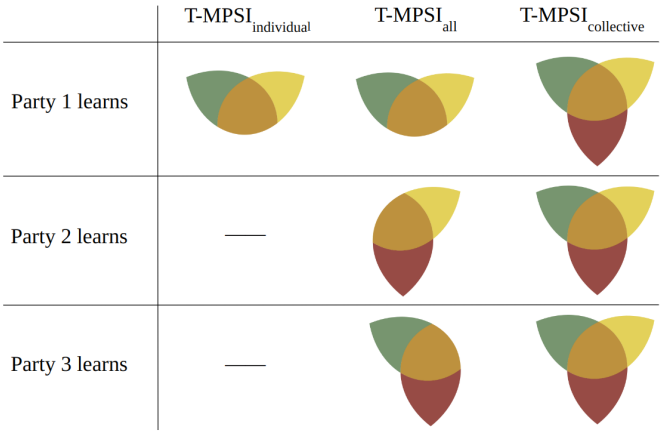

Fig. 2. Perspectives of T-MPSI with number of parties $m = 3$ and intersection threshold $T = 2$.

the union [16]. An MPSU protocol begins with each party encrypting their respective sets and exchanging them over secure channels. The sets are securely combined using privacy-preserving techniques, and finally, the aggregated encrypted union is decrypted to reveal the union of all individual sets.

C. Cryptographic Tools

The executions of MPSI and MPSU typically rely on encryption schemes and other cryptographic techniques detailed as follows [3], [11], [15], [17], [18].

Homomorphic encryption enables computations to be carried out on ciphertexts, yielding encrypted results that, when decrypted, correspond to the output of operations performed on the plaintexts. This characteristic ensures the processing of confidential information without giving access to the underlying data. More specifically, partially homomorphic encryption allows unlimited operations, but only of a single kind. For instance, additive homomorphic encryption supports the addition operation on ciphertexts. The Paillier cryptosystem is a type of public key encryption known for its additive homomorphic properties [19].

Secret sharing is a method for distributing a secret across multiple parties. The original secret can be reconstructed only when a sufficient threshold number of shares are combined. Below this threshold, no information about the secret can be learned. In particular, Shamir's secret sharing [20] is a commonly used secret sharing scheme based on the mathematical concept of polynomial interpolation, specifically Lagrange interpolation.

Oblivious transfer is a cryptographic protocol that enables a sender to transmit one of potentially many pieces of information to a receiver, without knowing which specific piece the receiver obtained [21]. The most common form, 1-out-of-2 oblivious transfer, involves the sender having two messages and the receiver wishing to learn one of them. The sender remains oblivious to which message the receiver chooses while ensuring that the receiver can only learn one of the messages.

III. Related Work

We now discuss relevant T-MPSI proposals, focusing on those that operate under the semi-honest security model. The semi-honest security model is also referred to as "honest-but-curious" or "passive" security. In this security model, adversaries might collude to attempt to learn as much information as they can, i.e., curious, but they do so while following the protocol faithfully, i.e., honest [22], [23].

Kissner and Song [15] were the first to propose a T-MPSI protocol, offering both T-MPSI$_{\text{all}}$ and T-MPSI$_{\text{collective}}$ variations, denoted as "threshold contribution" and "perfect threshold" respectively in their work. Their schemes employ polynomials and additively homomorphic encryption to securely compute the intersection, ensuring security against semi-honest adversaries. Designed for a fully connected network, this protocol exhibits a communication complexity that is cubic in the number of parties m. They do not provide an implementation for their T-MPSI schemes.

Mahdavi et al. [9] develop a T-MPSI$_{\text{all}}$ construction, focusing on enhancing security and efficiency through the use of oblivious pseudorandom functions, hashing, and additively homomorphic encryption. Assuming semi-honest behaviour among parties, the protocol can handle collusion of up to $m-1$ corrupt parties, where m is the number of parties in total. The authors claim that their protocol is able to handle large datasets, with communication complexity primarily dependent on the number of parties, the number of elements per set and the intersection threshold. The implementation of their T-MPSI protocol is available on GitHub[1].

Chandran et al. [11] introduce a T-MPSI$_{\text{individual}}$ scheme based on cuckoo hashing and secret sharing. It is designed to be secure under the semi-honest model with an honest majority, i.e., $\lfloor \frac{m-1}{2} \rfloor$. It operates efficiently under a star network topology, balancing security and computational demands across a server-centric system. Their current implementation[2] is broken. There are open issues regarding adding subdirectories and linking libraries. Therefore, we also exclude them from further analysis.

Bay et al. [3] propose a T-MPSI$_{\text{individual}}$ protocol leveraging Bloom filters and threshold homomorphic encryption. Their approach is tailored for scenarios involving a large number of participants with relatively smaller set sizes. Similarly to the scheme by presented by Chandran et al. [11], this protocol also operates under a star topology. It is secure in the semi-honest model for up to $m-1$ colluding parties. Their implementation is also available on GitHub[3].

IV. Collective T-MPSI

We now present our extension to transform any T-MPSI$_{\text{individual}}$ scheme [3], [11] into a T-MPSI$_{\text{collective}}$ scheme [15]. We consider only the semi-honest security model in our work. Although this model provides a weaker security

[1]https://github.com/cryspuwaterloo/OT-MP-PSI
[2]https://github.com/shahakash28/PQC-mPSI
[3]https://github.com/jellevos/threshold-multiparty-psi

guarantee than the malicious model, it still ensures no leakage of sensitive information, which aligns with our requirements. Furthermore, in collaborative environments like CTI sharing, the incentives for malicious behaviour are lower, making the semi-honest model more practical. Future work could extend the protocol to the malicious model using commitment schemes and zero-knowledge proofs.

A. Protocol Description

In a T-MPSI$_{\text{individual}}$ type construction, one party receives the intersection of their set with a threshold number $T-1$ of the other $m-1$ sets. In MPSI$_{\text{collective}}$, every party receives the threshold intersection of everyone's sets, i.e., all elements which appear in at least T of the m sets are outputted to all parties even if they do not possess the element.

We see that to convert T-MPSI$_{\text{individual}}$ into T-MPSI$_{\text{collective}}$, we first execute the T-MPSI$_{\text{individual}}$ protocol m times so that every party i has a turn to be the leader to obtain their T-MPSI$_{\text{individual}}$ result. This corresponds to T-MPSI$_{\text{all}}$. Then, we combine these m T-MPSI$_{\text{individual}}$ outputs in a privacy-preserving manner. For that, we use MPSU. An MPSU scheme is designed to ensure that no additional information beyond the union of the sets is leaked. This means parties do not learn whether a specific element from their set is included in someone else's set unless it is part of the union, thus maintaining the privacy of set contents. The final result is equivalent to the T-MPSI$_{\text{collective}}$ outcome (see Figure 3).

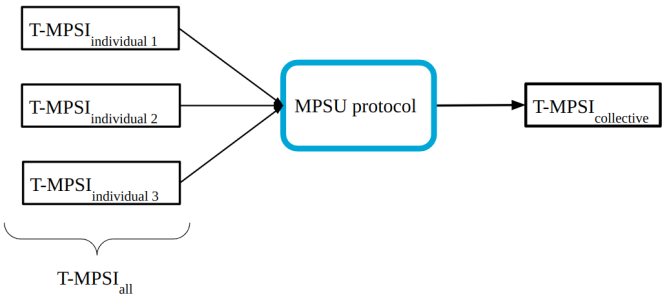

Fig. 3. Pipeline to transform a T-MPSI$_{\text{individual}}$ protocol into a T-MPSI$_{\text{collective}}$ protocol. In this example, there are three parties and sets involved.

Note that since we are pipelining two protocols together, the intermediate results of T-MPSI$_{\text{individual}}$ protocols, R_i, reveal the output of the T-MPSI$_{\text{individual}}$ in plaintext to each party i. However, this leakage does not compromise the privacy of the overall protocol since, given the end result R and the party's own private set S_i, R_i can be derived anyway (see the proof in Appendix A). The leakage is restricted to each individual party only, so each party i only sees the results relevant to their own data inputs. This means there is no additional risk of one party learning confidential details about another party's private set beyond what is jointly agreed to be shared.

B. Security Against Collusion

Depending on the values of the number of parties m, the intersection threshold T and the collusion threshold c, there could be information leakage. Consider the following scenario with $m=7$, $T=4$ and $c=3$. This means that all elements present in at least four of the seven sets will be part of the intersection. There are seven parties participating in the protocol, three of which may collude. The three colluding parties share their private inputs with one another to determine what elements the four possess. If neither of the three colluding parties possesses element e_1, but the intersection output contains e_1, then they conclude that the four non-colluding members must possess it. Therefore, we require that when using the T-MPSI$_{\text{collective}}$ protocol in practice, the user set $T > c$ and $m > T + c$.

V. Implementation

To the best of our knowledge, no implementation exists to convert between types of T-MPSI protocols. We present a generalized extension which can be applied to any T-MPSI$_{\text{individual}}$ scheme to transform it into a T-MPSI$_{\text{collective}}$ scheme.

A. Setup

In this work, we provide an example of such a pipeline using [3]'s implementation as the initial T-MPSI$_{\text{individual}}$ scheme. The first step is to convert it into a T-MPSI$_{\text{all}}$ scheme since the original code is designed so that only one party—the leader—learns which elements in their set do they share with the other parties. We do so by running the protocol concurrently, allowing every party to play the role of leader once to compute their T-MPSI$_{\text{individual}}$. If we were starting with a T-MPSI$_{\text{all}}$ protocol instead of T-MPSI$_{\text{individual}}$, we would not need to add this parallelization step since the T-MPSI$_{\text{all}}$ scheme already guarantees that every party obtains their T-MPSI$_{\text{individual}}$ set. The outputs from this first step are the T-MPSI$_{\text{individual}}$ intersection sets for each party. We store these intermediate intersections in JSON files, one file per party and one shared element per line. Then, we apply the MPSU protocol by Vos et al. [18] on these individual intersections. The resulting union represents the T-MPSI$_{\text{collective}}$ set. Vos et al.'s implementation provides users with the option to switch to MPSU large by specifying the `divisions` argument to 2. The MPSU large version of their protocol caters to when the universe size is much larger than the set sizes by reducing the number of required operations through strategic partitioning and selective processing. The MPSU protocol requires knowing the universe size prior to computing the private set union. In our experiments, we assume that the universe is all IPv4 addresses, therefore, the universal set size $|\mathcal{U}| = 2^{32}$ IPv4 addresses $- 588,514,304$ reserved addresses $= 3,706,452,992$ addresses.

We execute our runtime experiments on a laptop configured with a 64-bit Ubuntu 22.04.4 LTS operating system. The hardware includes an AMD Ryzen 9 6900HS processor with eight cores operating at a maximum frequency of 4.935 GHz as well as 24GB of RAM.

B. Parameters

We define the following parameters for our experiments.

- Number of parties: m = 5, 7, 9
- Number of elements per set: n = 128, 256, 512, 1024
- Intersection threshold: $T = \lfloor \frac{m+1}{2} \rfloor$
- Key length: $\kappa = 1024$
- Homomorphic PKE scheme threshold (for T-MSPI$_{\text{individual}}$): $\ell = \lfloor \frac{m}{2} \rfloor$
- Universe size (for MPSU): $|\mathcal{U}| = 3,706,452,992$ available IP addresses

The parameters were chosen to represent realistic values for CTI. The choices of m, T and κ are within the range commonly used in previous papers [11], [3]. The values of n used in this study are relatively low for a realistic scenario due to the limitations of the T-MPSI protocol and our hardware configuration. Larger n values would render the experiments impractically time-consuming on our laptop with limited RAM and CPU.

We set T to $\lfloor (m+1)/2 \rfloor$ for two reasons: to prioritize threats targeting at least half of the parties, which pose the greatest risk, and to ensure that attack details are shared only when a majority of group members are affected, preventing exposure of a minority within the group.

C. Runtime Performance

TABLE I
T-MPSI$_{\text{COLLECTIVE}}$ WITH THRESHOLD $T = \lfloor \frac{m+1}{2} \rfloor$ - RUNTIME IN SECONDS. THE VALUE BEFORE THE '+' REPRESENTS THE TIME FOR THE FIRST STEP OF COMPUTING THE T-MPSI$_{\text{ALL}}$ AND THE VALUE AFTER REPRESENTS THE TIME FOR THE SECOND STEP OF COMPUTING THE MPSU.

Max elements per set n	Number of parties m		
	5	7	9
128	159 + 4.63	366 + 5.58	669 + 5.89
256	314 + 4.67	709 + 5.82	1352 + 6.20
512	642 + 5.14	1424 + 6.26	2615 + 7.49
1024	1324 + 5.32	2973 + 6.81	5495 + 7.60

As demonstrated by Table I, for all tested values of m and n, the majority of the execution time occurs during the first step, i.e., calculating the T-MPSI$_{\text{all}}$. This involves running a T-MPSI$_{\text{individual}}$ protocol for all parties in parallel. In contrast, the time required for the second step to transition from T-MPSI$_{\text{all}}$ to T-MPSI$_{\text{collective}}$ by calculating the MPSU on the intermediate T-MPSI$_{\text{all}}$ results is minor.

This is because the runtime complexity of the T-MPSI and MPSU algorithms employed scale with the size of the party's sets [3], [18]. In the first step, transitioning from T-MPSI$_{\text{individual}}$ to T-MPSI$_{\text{all}}$, the input sets are large, up to $n = 1024$. However, the intersection sets outputted from the first step to be used as inputs for the second step are comparatively much smaller, with approximately 10 elements per set. Therefore, the MPSU protocol from the second step is applied on smaller set sizes.

Furthermore, from our experiments, we observe that the second step scales better than the first, staying below 10 seconds for all test cases, whereas the runtime for computing T-MPSI$_{\text{all}}$ grows rapidly as the number of parties is increased.

Overall, the first step of executing the T-MPSI$_{\text{all}}$ is still not efficient enough for practical use considering the scalability and performance requirements of the CTI sharing. CTI datasets often contain millions of entries, which translates to millions of elements per set, yet we observe slow runtimes starting at thousands of elements. Unless all participating organizations are equipped with powerful computers, it would be difficult to deploy the overall TMPSI$_{\text{collective}}$ implementation in a large-scale setting. Therefore, improvements in our extension rely on improvements in T-MPSI.

Even with these limitations, our results are promising since they indicate that our main idea of applying an MPSU protocol on the output of an T-MPSI$_{\text{all}}$ protocol to extend it to a T-MPSI$_{\text{collective}}$ protocol can be done efficiently. We believe the current selection of parameters aptly demonstrates the pipelines potential with realistic CTI sharing setups.

VI. CONCLUSION

In conclusion, our study highlights the challenges in researching and implementing T-MPSI protocols due to the conflation of various T-MPSI schemes. To provide more clarity, we defined three perspectives within T-MPSI: individual, all and collective. Although many implementations for T-MPSI$_{\text{individual}}$ and T-MPSI$_{\text{all}}$ exist, there is a lack of solutions that extend to T-MPSI$_{\text{collective}}$.

Our work addresses this gap by proposing a generic composition to convert any T-MPSI$_{\text{individual}}$ protocol into a T-MPSI$_{\text{collective}}$ protocol using MPSU, which aggregates the intermediate outputs from the T-MPSI$_{\text{all}}$ protocol. While our runtime experiments indicate that the extension holds potential, the scalability and performance of the initial T-MPSI$_{\text{individual}}$ step still fall short of the practical demands for CTI sharing.

Consequently, further improvement in the efficiency of T-MPSI is essential for developing robust and scalable solutions for secure data sharing.

REFERENCES

[1] V. Kolesnikov, N. Matania, B. Pinkas, M. Rosulek, and N. Trieu, "Practical multi-party private set intersection from symmetric-key techniques," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, B. Thuraisingham, D. Evans, T. Malkin, and D. Xu, Eds. ACM, 2017, pp. 1257–1272. [Online]. Available: https://doi.org/10.1145/3133956.3134065

[2] B. Pinkas, T. Schneider, and M. Zohner, "Scalable private set intersection based on OT extension," *ACM Trans. Priv. Secur.*, vol. 21, no. 2, pp. 7:1–7:35, 2018. [Online]. Available: https://doi.org/10.1145/3154794

[3] A. Bay, Z. Erkin, J. Hoepman, S. Samardjiska, and J. Vos, "Practical multi-party private set intersection protocols," *IEEE Trans. Inf. Forensics Secur.*, vol. 17, pp. 1–15, 2022. [Online]. Available: https://doi.org/10.1109/TIFS.2021.3118879

[4] A. Ben-Efraim, O. Nissenbaum, E. Omri, and A. Paskin-Cherniavsky, "Psimple: Practical multiparty maliciously-secure private set intersection," in *ASIA CCS '22: ACM Asia Conference on Computer and Communications Security, Nagasaki, Japan, 30 May 2022 - 3 June 2022*, Y. Suga, K. Sakurai, X. Ding, and K. Sako, Eds. ACM, 2022, pp. 1098–1112. [Online]. Available: https://doi.org/10.1145/3488932.3523254

[5] C. Beaman, A. Barkworth, T. D. Akande, S. Hakak, and M. K. Khan, "Ransomware: Recent advances, analysis, challenges and future research directions," *Comput. Secur.*, vol. 111, p. 102490, 2021. [Online]. Available: https://doi.org/10.1016/j.cose.2021.102490

[6] A. J. Burns, M. E. Johnson, and D. D. Caputo, "Spear phishing in a barrel: Insights from a targeted phishing campaign," *J. Organ. Comput. Electron. Commer.*, vol. 29, no. 1, pp. 24–39, 2019. [Online]. Available: https://doi.org/10.1080/10919392.2019.1552745

[7] NICCS. Cyber threat intelligence. [Online]. Available: https://niccs.cisa.gov/education-training/catalog/sans-institute/cyber-threat-intelligence

[8] IBM. What is threat intelligence? [Online]. Available: https://www.ibm.com/topics/threat-intelligence

[9] R. A. Mahdavi, T. Humphries, B. Kacsmar, S. Krastnikov, N. Lukas, J. A. Premkumar, M. Shafieinejad, S. Oya, F. Kerschbaum, and E. Blass, "Practical over-threshold multi-party private set intersection," in *ACSAC '20: Annual Computer Security Applications Conference, Virtual Event / Austin, TX, USA, 7-11 December, 2020*. ACM, 2020, pp. 772–783. [Online]. Available: https://doi.org/10.1145/3427228.3427267

[10] A. Miyaji and S. Nishida, "A scalable multiparty private set intersection," in *Network and System Security - 9th International Conference, NSS 2015, New York, NY, USA, November 3-5, 2015, Proceedings*, ser. Lecture Notes in Computer Science, M. Qiu, S. Xu, M. Yung, and H. Zhang, Eds., vol. 9408. Springer, 2015, pp. 376–385. [Online]. Available: https://doi.org/10.1007/978-3-319-25645-0_26

[11] N. Chandran, N. Dasgupta, D. Gupta, S. L. B. Obbattu, S. Sekar, and A. Shah, "Efficient linear multiparty PSI and extensions to circuit/quorum PSI," in *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, Y. Kim, J. Kim, G. Vigna, and E. Shi, Eds. ACM, 2021, pp. 1182–1204. [Online]. Available: https://doi.org/10.1145/3460120.3484591

[12] X. Yu, F. Li, W. Zhao, Z. Dai, and D. Tang, "Multiparty threshold private set intersection protocol with low communication complexity," vol. 2022, pp. 1–12, 2022. [Online]. Available: https://www.hindawi.com/journals/scn/2022/9245516/

[13] F. Liu, E. Zhang, and L. Qin, "Efficient multiparty probabilistic threshold private set intersection," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, CCS 2023, Copenhagen, Denmark, November 26-30, 2023*, W. Meng, C. D. Jensen, C. Cremers, and E. Kirda, Eds. ACM, 2023, pp. 2188–2201. [Online]. Available: https://doi.org/10.1145/3576915.3623158

[14] S. Ghosh and M. Simkin, "The communication complexity of threshold private set intersection," in *Advances in Cryptology - CRYPTO 2019 - 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2019, Proceedings, Part II*, ser. Lecture Notes in Computer Science, A. Boldyreva and D. Micciancio, Eds., vol. 11693. Springer, 2019, pp. 3–29. [Online]. Available: https://doi.org/10.1007/978-3-030-26951-7_1

[15] L. Kissner and D. Song, "Private and threshold set-intersection," 2004. [Online]. Available: http://www.dtic.mil/docs/citations/ADA461119

[16] J. Gao, S. Nguyen, and N. Trieu, "Toward A practical multi-party private set union," *IACR Cryptol. ePrint Arch.*, p. 1930, 2023. [Online]. Available: https://eprint.iacr.org/2023/1930

[17] M. Blanton and E. Aguiar, "Private and oblivious set and multiset operations," in *7th ACM Symposium on Information, Compuer and Communications Security, ASIACCS '12, Seoul, Korea, May 2-4, 2012*, H. Y. Youm and Y. Won, Eds. ACM, 2012, pp. 40–41. [Online]. Available: https://doi.org/10.1145/2414456.2414479

[18] J. Vos, M. Conti, and Z. Erkin, "Fast multi-party private set operations in the star topology from secure ands and ors," *IACR Cryptol. ePrint Arch.*, p. 721, 2022. [Online]. Available: https://eprint.iacr.org/2022/721

[19] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, ser. Lecture Notes in Computer Science, J. Stern, Ed., vol. 1592. Springer, 1999, pp. 223–238. [Online]. Available: https://doi.org/10.1007/3-540-48910-X_16

[20] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, 1979. [Online]. Available: https://doi.org/10.1145/359168.359176

[21] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Commun. ACM*, vol. 28, no. 6, pp. 637–647, 1985. [Online]. Available: https://doi.org/10.1145/3812.3818

[22] O. Goldreich, "Secure multi-party computation," *Manuscript. Preliminary version*, vol. 78, no. 110, 1998.

[23] Y. Lindell and B. Pinkas, "A proof of security of yao's protocol for two-party computation," *J. Cryptol.*, vol. 22, no. 2, pp. 161–188, 2009. [Online]. Available: https://doi.org/10.1007/s00145-008-9036-8

APPENDIX A
PROOF THAT INTERMEDIATE RESULTS CAN BE LEAKED

We prove that for all intersection threshold T and number of parties m that $S_i \cap R = R_i$. In other words, since the individual party can calculate the intermediate result from their own data and the collective output, the leakage does not expose any information that is not already accessible to them. Let R represent the union of intersections of all combinations of T sets out of the total m sets:

$$R = \bigcup_{\substack{J \subseteq \{1,\dots,m\} \\ |J|=T}} \bigcap_{j \in J} S_j. \tag{1}$$

Let R_i, for a specific party i, be defined as

$$R_i = S_i \cap \bigcup_{\substack{J \subseteq \{1,\dots,m\} \setminus \{i\} \\ |J|=T-1}} \bigcap_{j \in J} S_j. \tag{2}$$

We want to prove that we can obtain R_i from the intersection of S_i and R. We begin by expanding R into

$$S_i \cap R = S_i \cap \left(\bigcup_{\substack{J \subseteq \{1,\dots,m\} \\ |J|=T}} \bigcap_{j \in J} S_j \right). \tag{3}$$

Using the distributive property of intersection over union, this becomes

$$S_i \cap R = \bigcup_{\substack{J \subseteq \{1,\dots,m\} \\ |J|=T}} (S_i \cap \bigcap_{j \in J} S_j). \tag{4}$$

If we focus on subsets J such that $i \in J$, this reduces to considering intersections of $T-1$ other sets with S_i because the presence of S_i in the intersections is guaranteed:

$$S_i \cap R = \bigcup_{\substack{J \subseteq \{1,\dots,m\} \setminus \{i\} \\ |J|=T-1}} (S_i \cap \bigcap_{j \in J \cup \{i\}} S_j). \tag{5}$$

Since i is added back to J to form T elements, and J initially contains $T-1$ elements not including i, the equation reduces to

$$S_i \cap R = \bigcup_{\substack{J \subseteq \{1,\dots,m\} \setminus \{i\} \\ |J|=T-1}} (S_i \cap \bigcap_{j \in J} S_j). \tag{6}$$

By the distributive property of sets, we can extract the intersection of S_i out of the union, resulting in

$$S_i \cap R = S_i \cap \bigcup_{\substack{J \subseteq \{1,\dots,m\} \setminus \{i\} \\ |J|=T-1}} \bigcap_{j \in J} S_j. \tag{7}$$

This is exactly the definition of R_i as given by equation 2, so we conclude that

$$S_i \cap R = R_i. \tag{8}$$