Efficient Cloud-Assisted Protocol for Private Set Intersection Computation

Jihan Qin

School of Software (School of Cyberspace Security), Nanchang University, Nanchang, 330047, China

8008121147@email.ncu.edu.cn

Abstract. As cloud computing capabilities have advanced, coupled with the rise of cloud-assisted computing, innovative solutions for private set intersection computation have emerged. Traditionally, protocols designed for private data security entail extensive computational demands. This can be a burden for those involved in the privacy set seeking intersection. In many real-world scenarios, some participants might not possess the necessary computational power mandated by these protocols, leading to potential data vulnerabilities. However, the advent of cloud-assisted computing has introduced a transformative approach. Participants, rather than being limited by their local computational constraints, can now offload tasks to a cloud server. The server, equipped with vastly superior computing resources, undertakes the computation and subsequently returns the results. This not only alleviates the pressure on individual participants but also makes the process more adaptable to varied scenarios. In the discussed paper, a protocol is presented that adeptly harnesses cloud assistance to facilitate efficient private set intersection. This protocol not only ensures that data remains secure but also broadens the range of potential applications by accommodating different computational capacities. The marriage of cloud resources and private set computation offers a promising direction for the future of data security in a cloud-dominated era.

Keywords: Private set intersection (PSI), Cloud-assisted computation, Security mul- tiply computation, PSI protocol.

1. Introduction

Addressing the "data island" challenge is imperative in today's rapidly evolving landscape, where large-scale data technology, cloud computing adoption, and heightened awareness of secure information exchange converge. In our data-driven era, safeguarding against data breaches and misuse, shielding private information, and ensuring that data is both transmitted and computed securely are of paramount importance. Numerous countries worldwide have instituted relevant regulations and laws. These are designed to guide the appropriate use of data and to ensure its utilization in a manner that's both private and secure, thereby optimizing its potential benefits.

Private set intersection computation stands as a pivotal subfield within the domain of secure multi-party computation, boasting vast and significant real-world applications. This concept entails that each user holds a private set, and all users collectively determine the intersecting elements within these sets. Notably, during this process, no additional information is unveiled, ensuring that users remain unaware of each other's private sets. This idea finds its roots in Yao's renowned Millionaire Problem. In the realm of cloud-assisted computing, a client transmits a substantial dataset to a cloud server for swift and efficient processing. Upon completion of the computation, the server returns the results to the client. This dynamic capitalizes on the full computational prowess of cloud servers. Thus, private set intersection computation can seamlessly integrate with cloud assistance. Here, a user's private set is either obfuscated or encrypted before being sent to the cloud server, which then determines the intersection. This ensures the private data remains shielded from other users and also prevents potentially unreliable cloud servers from accessing the user's confidential information. Many contemporary cloud-assisted private set intersection computation protocols have been designed primarily for two-participant scenarios. While these offer commendable security and efficiency, extending them directly to multi-participant settings often proves challenging. Hence, this paper aims

to introduce a cloud-assisted protocol optimized for private set intersection computation involving multiple participants.

2. Related Work

2.1. Traditional Methods of Private Set Intersection

Traditional private set intersection methods often have commendable efficiency. For instance, straightforward hash solutions involve applying a hash function to set elements and then intersecting the resulting hash values to derive an answer. In their research, Huang et al. introduced an innovative approach to executing this protocol. They constructed a two-dimensional hash table row by row, which not only considerably reduced the number of security comparisons but also somewhat mitigated the inherent vulnerability of the prior method to collision attacks. Some researchers have also pivoted from a standard hash to the cuckoo hash in constructing the PSI protocol. This adaptation has proven effective in diminishing both the number of security comparisons and communication complexity [1]. Numerous scholars have ventured to develop PSI protocols through various other techniques, each with its unique strengths and weaknesses. One such method involves transforming the aggregation problem using secret sharing schemes. This strategy offers potential benefits in communication complexity and has garnered significant attention from the academic community. However, it's crucial to note that the PSI solutions discussed necessitate a certain computational prowess from all participating entities. If a participant's computational capacity is lacking, there's a significant risk that the private set intersection computation may not be completed securely.

2.2. Existing Cloud-Assisted Approaches

With the advent of cloud computing, corresponding PSI protocols that lean on cloud-assisted computing have emerged. Yet, based on current research findings, there remains ample opportunity for refinement and enhancement. Several existing protocols have been identified as having security vulnerabilities. Moreover, a harmonious balance between computational complexity, communication complexity, and security is still elusive. Some cloud-assisted PSI schemes, during the cloud-assisted computation phase, might unintentionally expose private or extraneous information. Such disclosures could pave the way for inference attacks, potentially unmasking sensitive data. While certain cloud-assisted PSI solutions boast commendable security measures, they may not be viable for clients with limited computational capacities. Such clients might struggle to execute the prescribed computational tasks locally, jeopardizing the privacy of their data. In one study, Shundong Li and colleagues presented a method to compute the intersection and concatenation of multiple private sets under a semi-honest model using Gödel coding. However, their method was tailored only for scenarios involving three participants or more [2]. Another study introduced a protocol built upon homomorphic encryption techniques, but it was marred by its extensive computational demands. Some protocols are either tailored exclusively for two participants or aren't scalable to accommodate multiple participants. Additionally, certain protocols grapple with fulfilling the computational demands of less capable clients without compromising security [3]. In essence, many of the current cloud-assisted PSI protocols fall short in efficiently delegating computational tasks to cloud servers while simultaneously ensuring robust security.

3. Preliminaries

3.1. Basic Concepts and Definitions

The field of contemporary cryptography emphasizes secret sharing, an indispensable tool for ensuring data confidentiality and information security. This method enables the distribution of a secret among a group of participants, primarily aiming to protect private data. The essence of secret sharing is the division of a secret into distinct shares, each held by different individuals. Only when

several participants collaborate can the secret message be reconstructed. Hence, a single participant alone cannot decipher the message. By ensuring that secrets aren't overly centralized, secret sharing can effectively thwart external attacks and betrayals from within the group. The methodology employed in dividing the secret and the subsequent recovery process are vital to the effectiveness of secret sharing. Shamir and Blackly first introduced the Shamir key sharing algorithm, leveraging vector methods and Lagrange interpolation. The fundamental idea here is that the sender divides a ciphertext, denoted as 's', into 't' shared units using a polynomial. This design ensures that any t units can collaboratively reconstruct s, but any t-1 of them remain clueless about s. The underpinnings of this key-sharing algorithm rest on Lagrange interpolation and vector methodologies. Homomorphic encryption is a cutting-edge cryptographic procedure anchored in intricate mathematical conundrums. When data undergoes processing via homomorphic encryption, it's operated upon in its encrypted state. Subsequent decryption of the operation's outcome yields a result identical to what would have been achieved had the operation been conducted on the original, unencrypted data. To put it succinctly, homomorphic encryption allows for computations on encrypted data without compromising result accuracy. This means one can outsource data processing to potentially untrustworthy entities without risking data exposure. In the realm of cryptographic functions boasting homomorphic properties, two plaintexts, 'a' and 'b', satisfy.

$$\mathrm{Dec}\,(\mathrm{En}\,(a)\ \odot\ \mathrm{En}\,(b)) = a\ \oplus\ b \tag{1}$$

The terms En and Dec stand for encryption and decryption operations, respectively. $\odot$ is the operation on the ciphertext and $\oplus$ is the operation on the plaintext. The encryption is referred to as additive homomorphic if $\oplus$ stands for addition. The encryption is referred to as multiplicative homomorphic encryption when $\oplus$ stands for multiplication. When both methods are utilized, the encryption is called fully homomorphic encryption.

3.2. Problem Statement

The cloud-assisted private set intersection computation problem can be defined as there exist n participants ($P1, P2, \cdots, Pn$) and each participant owns the privacy set Xi , that is, the private sets are $X1, X2, \cdots, Xn$. Then the requirement is $X1 \cap X2 \cap \cdots \cap Xn$, and it is desired to compute the intersection of these sets via an untrustworthy cloud server and return the result of the computation to the participants, without revealing any private data of the participants or any irrelevant additional data during the whole process.

4. Protocol

4.1. Protocol Overview

This section provides an overview of several cloud-assisted private set intersection computation protocols from recent literature. Firstly, articles [4] and [5] discuss the semi-trusted cloud-assisted private set intersection computation protocol, referred to as the basic protocol, and the semi-trusted cloud server-assisted private set intersection computation protocol, termed the full protocol. Both protocols from article [4] are designed specifically for two-participant computations, while the methods proposed in article [5] are versatile enough to accommodate both two and multi-participant scenarios. In the basic protocol from article [6], four entities are involved: the client (C), the service provider (S), the cloud server (H), and an oblivious two-party distributed pseudorandom function (Otd-PRF). Initially, the client C inputs a private set X, while the service provider S inputs a private set Y. This protocol comprises four phases: initialization, data outsourcing, cloud server computation, and client intersection retrieval. During the initialization, both the client C and the service provider S lay the groundwork via secret sharing. In the data outsourcing phase, both the service provider S and the cloud server H acquire encrypted sets. Following this, the cloud server H and the service provider S engage in the Otd-PRF to produce numerous instances. Through this computation, they derive a

new set R, which is subsequently sent to client C. Upon receiving set R, the client proceeds to validate and pinpoint the intersection elements. The full protocol, as presented in article [7], builds upon the basic protocol but integrates optimization strategies inspired by article [8], which utilizes the Cuckoo Hash, plain hash, and point set packing methodologies. Though the full protocol also follows the same four-phase structure as its basic counterpart, it incorporates additional tools such as a one-way hash function, the cuckoo hash algorithm, plain hash algorithm, and pack() and unpack() functions. A notable distinction in this full protocol is the employment of the cuckoo hash algorithm during the data outsourcing phase and the use of the plain hash algorithm in tandem with pack() and unpack() functions in the server computation phase. Consequently, there's a substantial reduction in both computational and communication complexities. Lastly, article [5] introduces a cloud server-assisted private set intersection protocol designed for multi-participant scenarios. This protocol harnesses the power of secret sharing, oblivious pseudorandom functions (OPRF), hash mapping algorithms, and key-value pair packing techniques. Under this scheme, it's posited that there are n participants, each possessing a unique private set Xi. One participant, denoted as Ci, is designated as the organizer. Upon the conclusion of the protocol, the organizer Ci obtains the computed results from the cloud server and reveals the set intersection to all participants.

4.2. Security Analysis

Computational complexity and communication complexity are often used as generic evaluation metrics for private set intersection computation protocols. In different protocols, the actual effectiveness of a protocol is measured by comparing the data of the two. In addition to this, whether the execution of the protocol prevents the cloud server from stealing the privacy collection, whether the data can be outsourced and whether the outsourcing of the computation serves a practical purpose are also important factors in measuring the goodness of the protocol. Both the basic protocol and the full protocol of the article are resistant to theft from semi-trusted cloud servers and are capable of efficient computational outsourcing, but neither enables data outsourcing, which can only be outsourced once and used once [9]. Overall, the protocol effectively implements a cloud-assisted scheme that performs private set intersection computation for both client and server participants. The protocols in the article also provide good results in terms of both computational complexity and communication complexity. Also, article has made a side-by-side comparison for different number of participants, and the experimental data are basically within the normal range [10]. On top of that, the protocol has no input domain limitations, no requirements for the number of cloud servers, and overall good results.

5. Conclusion

Private set intersection computation, especially within the realm of cloud-assisted environments, is of paramount importance in contemporary applications. The protocol delineated in this paper is not only fortified with robust security measures but also boasts remarkable efficiency under specific parameters. However, as with any pioneering endeavor, there's room for enhancement. One of the key strengths of this protocol is its emphasis on minimizing communication overhead. In the complex landscape of cloud interactions, where data is constantly exchanged between parties, reducing communication complexity can lead to significant performance improvements. It's not just about reducing the sheer volume of data transferred; it's about streamlining interactions to ensure that only essential data exchanges occur. Additionally, this protocol has been designed to judiciously lower the computational burden. In the era of rapid data processing, where time is of the essence, ensuring that computations are both swift and efficient is crucial. By focusing on lean computational methods, the protocol ensures that resources are optimally utilized, leading to faster results without compromising on accuracy. Perhaps one of the most intriguing facets of the protocol is its ability to enable data reuse by untrustworthy cloud servers. This feature addresses a vital challenge in the world of cloud computing: how can we harness the potential of vast data repositories without jeopardizing security?

By allowing for the reuse of data, the protocol ensures that the efficiency gains are maximized, without providing these servers with undue access or control over the data.

Reference

[1] Abadi, A., Terzis, S., & Dong, C. (2015). O-PSI: delegated private set intersection on outsourced datasets. In ICT Systems Security and Privacy Protection: 30th IFIP TC 11 International Conference, SEC 2015, Hamburg, Germany, May 26-28, 2015, Proceedings 30 (pp. 3-17). Springer International Publishing.

[2] Ruan, O., & Zeng, J. (2022). A delegated offline private set intersection protocol for cloud computing environments. In Proceedings of the 2022 2nd International Conference on Control and Intelligent Robotics (pp. 735-739).

[3] Ruan, O., Huang, X., & Mao, H. (2020). An efficient private set intersection protocol for the cloud computing environments. In 2020 IEEE 6th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS) (pp. 254-259). IEEE.

[4] Tian, H., Zhang, J., Yang, L., Tang, Y., & Zha, C. (2022). Multi-party privacy set intersection protocol with an untrusted cloud server. Frontiers in Computing and Intelligent Systems, 2(3), 68-74.

[5] Wei, L., Wang, Q., Zhang, L., Chen, C., Chen, Y., & Ning, J. (2022). Efficient privacy intersection computing protocol assisted by semi-trusted cloud services. Journal of Software, 34(2), 932-944.

[6] Li, S., Zhou, S., Guo, Y., Dou, J., & Wang, D. (2016). Aggregate privacy computing in cloud environments. Journal of Software, 27(6).

[7] Wang, Q., Wei, L., Liu, J., & Zhang, L. (2021). Multi-party privacy based on cloud server-assisted intersection computing protocol. Computer Science, 48(10), 301-307.

[8] Huang, Y., Evans, D., & Katz, J. (2012). Private set intersection: Are garbled circuits better than custom protocols?. In NDSS.

[9] Pinkas, B., Schneider, T., Segev, G., & Zohner, M. (2015). Phasing: Private set intersection using permutation-based hashing. In 24th USENIX Security Symposium (USENIX Security 15) (pp. 515-530).

[10] Kolesnikov, V., Matania, N., Pinkas, B., Rosulek, M., & Trieu, N. (2017). Practical multi-party private set intersection from symmetric-key techniques. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (pp. 1257-1272).