#300 Earn While You Reveal: Private Set Intersection that Rewards Participants

Email notification
Select to receive email on updates to reviews and comments.

▼ PC conflicts

Alberto Sonnino
Anna Maria Mandalari
Arthur Gervais
Ruba Abu-Salma

Rejected Round 2

Submission (3MB) 4 May 2023 12:30:29am AoE · 6c698973

▶ Abstract

In Private Set Intersection protocols (PSIs), a non-empty result always reveals something about the private input sets of the parties. Moreover, in various variants of PSI, not all parties necessarily receive or are interested in the result. Nevertheless. to date. the literature has

[more]

▶ Authors (anonymous)

A. Abadi, S. Murdoch [details]

Track

Applied Cryptography

▶ Topics

	DoePapRai	RevExpDom	RevCon	OveMer
Review #300A	1	1	2	2
Review #300B	1	3	3	3
Review #300C	1	3	3	3

1 Comment: *Rebuttal Response* (A. Abadi)

You are an **author** of this submission.

Edit submission Add comment

Reviews and comments in plain text

Review #300A

Paper summary

The paper presents a protocol, Justitia, for fair multi-party PSI, and built on that another protocol, Anesidora, for fair multi-party PSI with compensation (for privacy leakage).

Strengths / Reasons to accept

This is a powerful functionality. Justitia could probably be used to build many kinds of further interesting functionalities (beyond Anesidora). The paper is also quite detailed and clearly written, with a huge number of appendices giving further clarification on the many primitives used to build these protocols.

Weaknesses / Reasons to reject

- The protocols are pretty complex and operate under strong assumptions, building heavily on smart contracts. They also have somewhat complex set of malicious/semi-honest and non-collusion assumptions on the numerous parties involved. For example, in what situation do you find a Dealer who is supposed to be semi-honest to participate in Justitia? Anesidora has even more complex assumptions, including some rationality assumptions. I have a hard time believing that this is actually secure in any meaningful real-world scenario.
- The performance evaluation is only asymptotic. I would be great to see some concrete numbers, and an estimate of the gas cost in evaluating these smart contracts. Is this actually realistic to run for large sets? What does the gas cost mostly depend on?

Constructive comments for author

This seems pretty impressive, and the paper is nicely put together, but it's just quite complicated as a whole. Maybe some of these components could be abstracted out into bigger units that are easier to reason about for the reader. For a reader who is not very comfortable with thinking about the security of smart contracts, maybe you could include more explanation in the main body of the paper. Maybe having clearly written ideal functionality boxes for the SCs would help.

Questions for authors' response

- In the protocol descriptions, say 6.2 (Justitia), the system parameters are not mentioned explicitly or explained properly. For example, there are things like the table size, the field size, etc., which I guess need to be agreed upon, but are not mentioned anywhere. Their required sizes are not mentioned anywhere either. How big does the field need to be?
- In 6.2 step (2), what happens if this hashing process fails, i.e., the pre-defined bin size is insufficient? Does this leak information?
- In 6.2 step (5), it doesn't seem obvious that the polynomial omega^{C,D}*pi^{C} wouldn't have any zero coefficients. For example, if just one of the s_i' in (2) is zero, then this product will always have at least one zero coefficient. There doesn't seem to be any recovery strategy in case this happens.
- It's not entirely clear to me whether the polynomial zeta is secret, and from whom, or until what point.

Does the paper raise ethical concerns?

1. No

Reviewer expertise in this domain

1. No familiarity

Reviewer confidence

2. Somewhat confident

Overall merit

2. Weak reject

Review #300B

Paper summary

This paper tackles the problem of how parties are incentivized to contribute to private set intersection (PSI), especially if these parties do not get an output. To this end, the work proposes a new multi-party protocol with financial rewards for parties providing private sets as input ("Anesidora"). On this path, the first fair multi-party PSI ("Justitia") is presented and both

constructions are proven secure using a simulation-based argument. The work considers an active adversary corrupting the majority of clients under static corruption.

The protocols are based on a variety of building blocks. One of these is a new notion called unforgeable polynomials. This primitive allows to generate authenticated polynomials such that a verifier can check if the polynomial is intact.

The presented multi-party PSI protocol with financial rewards has similar computation and slightly higher communication complexity than other state-of-the-art multi-party PSI protocols. However, the paper is the only one providing financial rewards by integrating smart contracts into the protocol design.

Strengths / Reasons to accept

The paper tackles an interesting research question: how parties can be rewarded for contributing private information to the private set intersection computation? The paper is very well written and very self-containing. I like the additional details in the appendix on the building blocks. Also, the security analysis looks rigorous, although I did not check all the details.

Weaknesses / Reasons to reject

In my opinion, the major weakness is the lack of some intuition and explanation about the constructions up-front and about some design/modeling choices. I needed this intuition to see the big picture while reading the building blocks, especially since there are quite some preliminaries and subroutines. Additionally, some design/modeling choices are unclear to me, e.g., why did you model a designated dealer, how did you distinguish between passive and active corruption, and why did you use the set of predicates Q to define the notion of PSI with fair compensation?

Finally, I was missing a gas cost analysis for the smart contract to assess the practicality of the construction.

Constructive comments for author

For me, the most significant aspects of improving are the points listed under weaknesses, especially the intuition at the beginning. These changes would help reading the paper more easily.

The challenges stated in the introduction (p.1) seem very generic. For instance, keeping the overhead low seems challenging for most MPC protocols aiming for efficiency. I think your

constructions also have more specific challenges, which might be stated instead.

I need clarification on how the wrapped functionality works and why you need ω to construct an unforgeable polynomial. This could be explained.

Some minor comments:

- typo: "output of one subroutine is fed a[s] input into ..." (p.1)
- "such as Programmable Pseudorandom Function (OPPRF)", where does the O in the abbreviation come from? (p.2)
- It was unclear what "Del" stands for in Q^{Del} (Section 4) before looking into the appendix.

Questions for authors' response

- How did you distinguish between passive and active corruption, and how do you motivate considering passive corruption of the dealer and the auditor?
- How efficient is the smart contract execution in terms of gas costs?
- What is the running time of the entire Anesidora protocol?

Does the paper raise ethical concerns?

1. No

Concerns to be addressed during the revision/shepherding

- providing gas costs for deploying and running the smart contract
- providing a runtime analysis

Reviewer expertise in this domain

3. Knowledgeable (I don't necessarily work in this topic domain, but I work on related topics or I am cognizant of the work in this domain in recent years)

Reviewer confidence

3. Quite confident

Overall merit

3. Neutral

Review #300C

14 Jul 2023

Paper summary

This considers the problem of incentives to participate in a secure multi-party computation. Clearly, one can provide payments to the honest parties that contribute to the success of the MPC with their inputs. But how do we guarantee that the payment is delivered? The obvious approach would be to have a trusted party in charge of the payments. This paper weakens the assumption by having a smart contract on a blockchain deliver the payments. More specifically, this paper looks at the problem of Private Set Intersection, a special notable example of MPC, and considers two different notions and propose an efficient implementation for each of them: PSI with Fair Compensation $\mathcal{PSI}^{\mathcal{FC}}$ that honest parties either obtain the result or are financially compensated, if the computation aborts. PSI with Fair Compensation and Reward $\mathcal{PSI}^{\mathcal{FCR}}$ guarantees that honest are rewarded for their work and compensated in case of abort.

The core of the contribution is the design of PSI protocols for which dishonest parties can be identified and honest parties can submit a "proof" of their good behavior so that the smart contract can release the payment.

Strengths / Reasons to accept

Very interesting and meaningful problem

Weaknesses / Reasons to reject

The paper lacks an experimental part that shows that this approach is indeed practical on current blockchains.

Constructive comments for author

It would help the reader to clearly identify the needed capabilities of the smart contract and to compare them with what, say, Ethereum offers. This will provide a way to see how close the proposal is to be implementable on some real blockchain. Also, it would be very useful if you could point out why it is not a threat that the inputs to the smart contract are revealed.

Post-Rebuttal Summary

The reviewers agreed that the paper tackles an interesting and meaningful problem. The theoretical contribution of the paper consisting in putting forth a model with a weak third party implemented by a smart contract was appreciated. There was consensus among the reviewers felt though that, in absence of an experimental evaluation, it was difficult to evaluate the practicality of the proposed constructions if run on current blockchain (like Ethereum).

Questions for authors' response

1. How is the smart contract different from an extra honest but curios party that has an offline way to transfer monetary payments?
2. Does the smart contract need to monitor the execution of the protocol? In other words is it possible for a party to behave maliciously in the actual execution but then provide correct information to the smart contract so to obtain the payment?
3. What is the role of the smart contract in the ZSPA protocol of Figure 2 and Figure 3?
4. Since the paper mentions Ethereum, do we have an estimate of the gas cost for running the protocols?

Does the paper raise ethical concerns?

1. No

Reviewer expertise in this domain

3. Knowledgeable (I don't necessarily work in this topic domain, but I work on related topics or I am cognizant of the work in this domain in recent years)

Reviewer confidence

3. Quite confident

Overall merit

3. Neutral

Rebuttal Response

Author [Aydin Abadi] 28 Jun 2023 **1286 words**

We would like to thank the reviewers for their valuable comments.

Reviewer A

A.1. security assumption and semi-honest dealer:

We allow the majority of the parties (including the extractors) to be corrupted by active adversaries during the computation of PSI which enables Justitia/Anesidora to offer a strong security guarantee. We only require the dealer to be a semi-honest non-colluding adversary which is not a strong assumption. We also allow the two extractors to be rational colluding adversaries when it comes to extracting the final result and distributing the rewards. There exist various PSI protocols that have been published at ACM CCS and considered only semi-honest adversaries. For instance, those that have been proposed in [10, 33, a]. Also, the idea of efficiently delegating a computation to two potentially colluding rational servers was proposed in [16] and published at ACM CCS 2017.

A.2. The protocols' performance and the gas cost:

- Overall performance: We have left the implementation and concrete cost analysis of the proposed protocols for future work. However, we can estimate the cost of our protocol by comparing it with the PSI protocols in [1, b] that use similar techniques, in particular polynomial representation. Our protocols (i.e., Justitia and Anesidora) are more efficient than the PSIs in [1, b]. Because in our protocols the users do not need to factorise the final polynomials to find the intersection. Instead, the users find the intersection by simply evaluating their set elements at the final polynomials. However, in [1, b] since users do not have their set elements locally they need to factorise the final polynomials. The polynomial factorisation is costly and dominates other (symmetric-key-based) operations.

- Gas cost: In our protocols, the main gas cost stem from the polynomial division, which requires checking whether a polynomial (of degree 200) is divisible by a polynomial of degree 1, i.e., polynomial ζ in the protocol. This is done for every bin of the hash table. As it is shown in [b] when each party has about 1,000,000 set elements (or about 2^{20}), then the number of bins in the hash table is about 4,1943 and the number of elements in each bin can be set to 100. Thus, the total number of divisions that the smart contract must perform is only 4,1943 (when each party holds 2^{20} set elements). This cost is independent of the number of parties that are involved in the protocol.

A.3. System parameters:

The hash table's size has been studied before and we can use the parameters established in the literature, e.g., in [b, full version]. For instance, when each party has set elements of 2^{20} the number of bins can be set to 4,1943. Also, each bin's capacity is set to 100 (regardless of the total number of elements). The field size can be as small as 64-bit or 128-bit, depending on the set element size. We will clarify the above points in the paper.

A.4. What happens if this hashing process fails?:

As we stated in Appendix D (page 17), given the maxim number of set elements, we can set the hash table's parameters in such a way that a bin receives at most a predefined constant number of elements (i.e., d = 100) except with a negligible probability. Thus, the hashing process will not fail except with a negligible probability. Even if it fails, then there will be no leakage.

A.5. In 6.2 step (5), zero coefficients the polynomial $\omega^{C,D} \cdot \pi^{C}$:

As we stated in step (5) (page 8) each party ensures that the polynomial $\omega^{C,D} \cdot \pi^{C}$ does not have any zero coefficient (if it has, then the party simply picks another random polynomial $\omega^{C,D}$). Moreover, as we analysed in Appendix N.2 (page 22), even without such a check, with a negligible probability one of the coefficients of the polynomial $\omega^{C,D} \cdot \pi^{C}$ is zero.

A.6. Is the polynomial zeta secret?

The dealer is the only party who picks and knows the secret polynomial ζ. This polynomial remains secret until all parties provide their messages to the smart contract. Then, the dealer in step 10 sends the secret polynomial ζ to the smart contract. The last line in step 10 (page 8) explicitly states that the dealer sends ζ to the smart contract.

Reviewer B

B.1. Additionally, some design/modelling choices are unclear, distinguishing between passive and active corruption, and use the set of predicates Q to

define the notion of PSI with fair compensation:

- choosing a designated dealer: we used a designated dealer to keep the overall cost low. We have explained this point in detail in Appendix N.3.1 (page 23).
- passive and active corruption: we have formally defined what we mean by active/malicious and passive adversaries (in Section 3.1., page 2)– as the reviewers know, they are well-established concepts. Also, see our response to reviewer A.1.
- design choice: For every subroutine that we designed, we briefly explained why we use them in the protocol. For instance, for VOPR (section 5.1., page 5) we stated that: “We will use VOPR in Justitia for two main reasons...”.
- predicates: the standard simulation-based paradigm offers only (1) privacy in the semi-honest adversarial model and (2) privacy and output correctness in the active/malicious adversarial model. This paradigm does not offer additional security guarantees (e.g., fairness, or compensation) that some protocols may need. So, to capture additional security requirements, it is common to define additional predicates that offer those security guarantees. As we stated in Appendix H (page 18), some of these predicates were previously defined in [31].

B.2. gas cost analysis:

See our response to A.2

B.3. need clarification on how the wrapped functionality works and why you need ω to construct an unforgeable polynomial.

- Wrapped functionality: see our reply to B.1.
- ω: since in Justitia polynomial π must be multiplied by a random polynomial ω (for the basic security of PSI as described in Section 3.8, pages 3--4), we included ω in the unforgettable polynomial as well. However, in general, the unforgeable polynomial works and its proof holds, even without the involvement of ω.

Reviewer C

C.1. the needed capabilities of the smart contract:

Standard public (Ethereum) smart contracts meet our protocols' requirements. In our protocol smart contracts mainly (1) distribute deposits, (2) keep some values/state, and (3) divide polynomials by a polynomial of degree 1.

C.2. why is it not a threat that the inputs to the smart contract are revealed?

All plaintext messages (e.g., set elements or polynomials representing the sets) that are sent to the smart contract are encrypted before sending them to the smart contract. Also, see Appendix N.3.3 (page 23) for further discussion.

C.3. How is the smart contract different from an extra honest but curios party?

We do agree with the reviewer that (to achieve efficiency) we can replace the smart contracts with a semi-honest party (at the cost of making the protocols more centralised).

C.4. Does the smart contract need to monitor the execution of the protocol?

Our PSI protocols ensure that any party misbehaving during the execution of the (offline) subroutine protocols will be detected and identified. In this case, a malicious party cannot provide a valid message (i.e., a polynomial which contains factor ζ) to the smart contract, except for a negligible probability. Thus, by checking the correctness of the messages that the parties provide to the smart contract, the contract "monitors the execution" of the subroutine protocols.

C.5. What is the role of the smart contract in the ZSPA protocol of Figure 2 and Figure 3?

It acts only as a bulletin board.

References:

a. Chen, Hao, et al. "Fast private set intersection from homomorphic encryption." In CCS 2017.

b. Abadi, A., et al. Multi-party updatable delegated private set intersection. In FC 2022.--- (full version): https://fc22.ifca.ai/preproceedings/68.pdf