Security of Private Set Operation Schemes: Separations and Implications

Mojtaba Rafiee[1,2*]

[1*]Department of Applied Mathematics and Computer Science, Faculty of Mathematics and Statistics, University of Isfahan, Isfahan, Iran.
[2]School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran.

Corresponding author(s). E-mail(s): m.rafiee@mcs.ui.ac.ir;

Abstract

The private set operation (PSO) scheme [Rafiee-Khazaei, Comput. J. 2020] is a cryptographic primitive that enables a user to securely outsource their dataset to cloud server, and then when needed, securely issue common set operation queries to the server and receive the results. This primitive has always been of interest to researchers because it supports set operations, which are the most basic mathematical operations and are used in a wide range of real-world applications. In previous research, security notions such as: `naSIM` and `aIND` have been introduced for it. In this paper, we develop the standard security notions for **PSO** schemes: an adaptive version of simulation-based security notion (`aSIM`) and a non-adaptive version of indistinguishability-based security notion (`naIND`). We also study the relation between these security notions and determine their implications and separations. In addition to these, we also provide a summary of the available **PSO** constructions and their security level, and introduce research potentials in this regard.

Keywords: Security, Simulation, Indistinguishability, Separation, Implication

1 Introduction

The development of emerging technologies has caused the data volume to increase rapidly, and organizations are faced with management and maintenance challenges. One of the popular and well-known solutions to overcome these challenges is to use

outsourcing services in cloud computing platform. Due to the fact that data is stored on an external server and away from the data owner, using these services always brings security concerns [1, 2]. One of the most important security concerns is the privacy preserving of outsourced data [3]. In fact, the data owner wants to maintain the privacy of their data from unauthorized persons while enjoying the benefits of outsourcing. Data encryption before data outsourcing is an approach to ensure the privacy of the outsourced data. However, due to the variety of queries, appropriate encryption schemes must be used to extract desired results from outsourced data, and classical encryption schemes does not meet such conditions. In this regard, a wide variety of cryptographic primitives have been proposed, such as: Searchable Symmetric Encryption (SSE) schemes [4–6], Tuple Set (Tset) schemes [7–9], Adjustable Join (Adjoin) schemes [10–12], and Private Set Operation (PSO) schemes. However, among these schemes, the PSO schemes have always received special attention due to their support for set operations, which are known as the most basic mathematical operations and a large number of applications can be built based on them.

A PSO scheme is a cryptographic primitive that enables a user to securely outsource their dataset to cloud server, and then when needed, securely issue common set operation queries to the server and receive the results. For the first time, Rafiee et al. [13] have proposed the syntax and security notions of PSO schemes. In particular, they have presented a non-adaptive simulation-based security notion called naSIM. They have also introduced, for the first time, a reasonable leakage profile for these schemes. To demonstrate the validity of their security notions, they present a generic PSO construction based on the cryptographic primitives Tset and Adjoin. Subsequently, Rafiee in paper [14] has proposed a new adaptive indistinguishability-based security notion called aIND. They showed that their security notion is strictly stronger than naSIM security notion. Similar to previous research, they have proposed a generic PSO construction to demonstrate the validity of their security notion.

Motivation. In this paper, we develop standard simulation-based and indistinguishable-based security notions for PSO schemes. More precisely, we introduce an adaptive version of security notion naSIM, which we have named aSIM, as well as a non-adaptive version of security notion aIND, which we have named naIND. Since that the standard security notions provided for most schemes in the literature include four security notions: naIND, aIND, naSIM, aSIM, our new security definitions complete these security notions for PSO schemes. We also investigate the implications and separations between these security notions, and determine the security level covered by the available PSO constructions.

The results obtained from this paper can facilitate the process of: 1) providing new protocols based on available PSO schemes, 2) familiarity with standard security concepts to gain the ability to introduce new security concepts and identify their positions among other existing security notions, 3) introducing new efficient construction that satisfy these security notions.

1.1 Contributions

In this paper, we introduce two new security notions for PSO schemes: naIND and aSIM. Also, we study and prove the implications and separations between all security notions defined for PSO schemes. In the following, the general description of each of the above innovations is given.

Natural simulation-based and indistinguishability-based notions. We develop the natural indistinguishability-based and simulation-based security notions for PSO schemes by introducing two new security notions, naIND and aSIM. Figure 1 shows all of the PSO security notions that we study in this paper. See Sections 4 and 5 for more details on each security notion.

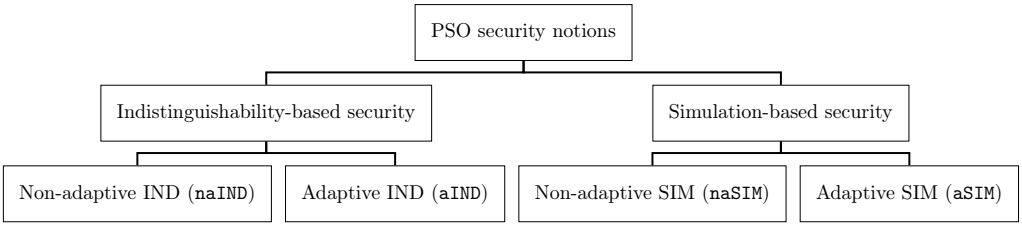

Fig. 1: The security notions of the PSO schemes.

Separations and implications. We have proved almost all the relations between the security notions proposed for PSO schemes. The results obtained can be seen in Figure 2. In this figure, the relation "$A \longrightarrow B$" indicates that the notion A implies the notion B, and the relation "$A \not\longrightarrow B$" indicates that the notion A does not necessarily imply the notion B.

The implications and separations between the security notions naSIM and aIND are studied in paper [14]. However, the rest of the relations that have been investigated and proven in this paper are as follows:

- We examined the relation between security notions naIND and aIND in Subsection 4.3 and showed that security notion aIND is strictly stronger than security notion naIND.
- We investigated the relation between security notions naSIM and aSIM in Subsection 5.3 and showed that security notion aSIM is strictly stronger than security notion naSIM.
- We studied the relation between naSIM and naIND in Section 6 and showed that these two security notions are equivalent.
- We also showed that security notion aSIM implies security notion aIND. However, the converse of this implication (i.e., aIND $\stackrel{?}{\longrightarrow}$ aSIM) remains as an open problem. See Section 6 for more details.

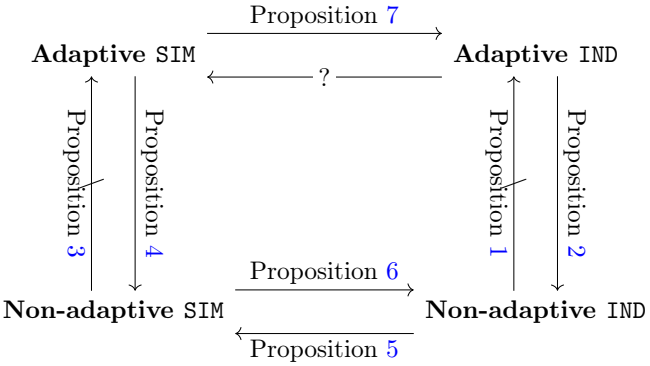

Fig. 2: Relations between PSO security notions.

1.2 Paper organization

The rest of this paper is organized as follows. In Section 2, we present the notations that are used throughout the paper, as well as the syntax and correctness of PSO schemes. We explain the PSO leakage function in Section 3. The indistinguishability-based and simulation- based security notions for PSO schemes are presented in Sections 4 and 5, respectively. The relations between these security notions are presented in Section 6. In Section 7, we provide a detailed overview of the available PSO constructions in terms of security and efficiency. Finally, Section 8 concludes the paper and points out future directions.

2 Preliminaries

In this section, we introduce the required background. In particular, we first present the notations used throughout the paper. Next, we review the syntax notion and correctness condition of PSO schemes.

2.1 Notations

Throughout the paper, we use λ to denote the security parameter. Let A is a probabilistic algorithm, we use the expression $y \leftarrow A(x)$ to denote that y is the output of A on input x. Let S is a finite set, $x \leftarrow S$ stands for uniformly selecting an element x from set S. We say that a function such as ε is negligible, if it is smaller than the inverse of any polynomial in security parameter for sufficiently large values of λ. As a convention, we consider the output of a security experiment by the experiment name itself.

2.2 PSO scheme

A private set operation scheme, first proposed by Rafiee et al. in [13], is a symmetric cryptographic primitive that enables a user to securely outsource his dataset to a

cloud server, and then when needed, securely issue common set queries to the server and receive their results. In the following, we first explain how to use this scheme and then review its formal definition.

The use of this scheme is as follows. The user first runs Algorithm KeyGen and obtains a set of public and private parameters that are used to securely outsource data and generate tokens. In the next step, the user encrypts each word in each set of his dataset using Algorithm Enc, and prepares the encrypted dataset $\widetilde{\mathcal{DS}}$ for outsourcing on the cloud server. Later, as soon as a set operation is needed, the user calls Algorithm Token to generate the desired token and sends it to the server. Upon receiving the token, the server computes the query result using Algorithm Operate and returns it to the user. In this step, the obtained results are in an encrypted format and the server does not get any information about its plain content. Finally, the user decrypts the results returned from the server using Algorithm Retrieve and obtains the plain result. Below, the formal definition of the scheme is given.

Definition 1 (PSO syntax [13]) A private set operation (PSO) scheme is a tuple of PPT algorithms PSO = (KeyGen, Enc, Token, Operate, Retrieve), where:

- $(Param, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$: It is a probabilistic key generation algorithm that gives a security parameter λ, and returns as output a secret key K and a set of public parameters $Param$.
- $(\tilde{l}, \tilde{w}) \leftarrow \mathsf{Enc}_K(l, w)$: It is a probabilistic encryption algorithm that takes as input a secret key K, a word w and a set label l, and returns as output an encrypted-word/set-label pair $(\tilde{l}, \tilde{w})$.
- $sot \leftarrow \mathsf{Token}_K(q)$: It is a probabilistic token generation algorithm that takes as input a secret key K and a set query q, and returns as output a set operation token sot.
- $\widetilde{S} \leftarrow \mathsf{Operate}_{Param}(\widetilde{\mathcal{DS}}, sot)$: It is a deterministic algorithm that takes as input a set of public parameters $Param$, an encrypted dataset $\widetilde{\mathcal{DS}}$ and a set operation token sot, and returns as output an encrypted set $\widetilde{S}$.
- $S \leftarrow \mathsf{Retrieve}_K(\widetilde{S})$: It is a deterministic algorithm that takes as input a secret key K and an encrypted set $\widetilde{S}$, and returns as output a result set S.

Correctness. The PSO correctness guarantees that no PPT adversary can find a query $q = (op, l_k, l_p)$ and a dataset $\mathcal{DS} = \left(l_i, (w_j)_{j=1}^{m_i}\right)_{i=1}^{n}$ such that:

$$\left(\mathcal{DS}[l_k] \; op \; \mathcal{DS}[l_p]\right) \neq \mathsf{Retrieve}_K(\mathsf{Operate}_{Param}(\widetilde{\mathcal{DS}}, sot)),$$

where $sot \leftarrow \mathsf{Token}_K(q)$, and for every $i \in n$ and $j \in m_i$ we have:

$$\widetilde{\mathcal{DS}}[l_i][w_j] \leftarrow \mathsf{Enc}_K(\mathcal{DS}[l_i][w_j], l_i).$$

3 PSO leakage function

The indistinguishability-based and simulation-based security notions, introduced in Sections 4 and 5, are parameterized based on a deterministic function Leak, called the leakage function. This function was first introduced by Rafiee et al. in [13], and models all reasonable information that is revealed when outsourcing and issuing future

set operation queries. The leakage function is used in the indistinguishability-based security notions to introduce a valid adversary. In fact, such an adversary, who is constrained by the leak function, is prohibited from obvious successes in security games, and thus a meaningful security definition becomes possible. On the other hand, the leak function in simulation-based security notions determines the information that is given to the simulator and it must interact with the adversary based on it. This interaction should be in such a way that the adversary does not recognize whether the interaction with him is through real data or a leakage profile. In the following, we review the leakage function of PSO schemes. To see the formal definition of this function, interested readers can refer to paper [13].

PSO leakage function. For a dataset $\mathcal{DS} = (l_i, S_i)_{i=1}^{n}$ and a list of set queries $\mathcal{Q} = (q_i)_{i=1}^{m}$, the minimal leakage function are defined as follows:

$$(\mathrm{Dim}, \mathrm{SP}, \mathrm{OPP}, \mathrm{RP}, \mathrm{DP}) \leftarrow \mathsf{Leak}(\mathcal{DS}, \mathcal{Q}),$$

where:

- **Dimension (Dim).** It is the dataset dimensions that determines the total number of sets (n) and the cardinality of every set ($\mid S_i \mid$),
- **Search pattern (SP).** It determines the repetition of retrieved sets in different queries,
- **OPeration pattern (OPP).** For each query issued, it determines the set operation,
- **Result pattern (RP).** For each query issued, it shows the result set,
- **Duplication pattern (DP).** For each set in the dataset, it determines the locations with the same content.

4 Indistinguishability-based security notions

In this section, we first study the indistinguishability-based security notions in both adaptive and non-adaptive versions. The adaptive version of this security notion was first presented in paper [14], and we review it for recall. However, we introduce its non-adaptive version for the first time, based on standard security notions in the literature [15]. Finally, we investigate the relation between these two security notions and show that the adaptive version is strictly stronger than its non-adaptive version.

4.1 naIND security

The non-adaptive indistinguishability-based security notion, which we have called naIND, is one of the simplest and most natural security notions. This security notion considers two entities: a challenger and an adversary. The challenger first generates public and private parameters of the PSO scheme and sends its public parameters to the adversary. Upon receiving the public parameters, the adversary selects two challenge datasets $\mathcal{DS}_0$ and $\mathcal{DS}_1$ and a list of challenge queries $\mathcal{Q}$ such that $\mathsf{Leak}(\mathcal{DS}_0, \mathcal{Q}) = \mathsf{Leak}(\mathcal{DS}_1, \mathcal{Q})$, and sends them to the challenger. An adversary who satisfies the above condition is called a valid adversary, and the security experiment

is only meaningful for such an adversary. In such a setting, this security notion guarantees that the pairs $\big(\mathsf{Enc}_K(\mathcal{DS}_0), \mathsf{Token}_K(\mathcal{Q})\big)$ and $\big(\mathsf{Enc}_K(\mathcal{DS}_1), \mathsf{Token}_K(\mathcal{Q})\big)$ generated by the challenger are computationally indistinguishable to the adversary. In the following, the formal definition of this security notion is given.

Definition 2 (naIND security) An PSO scheme such as $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Token}, \mathsf{Operate}, \mathsf{Retrieve})$ is called non-adaptively indistinguishable-based secure with respect to the leakage function Leak (abbreviated Leak-naIND-secure) if for every PPT valid adversary $\mathcal{A}$, there exists a negligible function ε such that

$$| \Pr[\mathsf{Exp}^{\mathsf{naIND}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1] - \Pr[\mathsf{Exp}^{\mathsf{naIND}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1] | \leq \varepsilon(\lambda),$$

where for each $b \in \{0, 1\}$, the experiment $\mathsf{Exp}^{\mathsf{naIND}}_{\Pi,\mathcal{A}}(\lambda, b)$ is defined as follows.

1. **Setup phase:** The challenger Chal samples $(Param, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and sends the public parameters $Param$ to the adversary $\mathcal{A}$.
2. **Challenge phase:** The adversary $\mathcal{A}$ chooses two dataset $\mathcal{DS}_0$, $\mathcal{DS}_1$, and a list of queries $\mathcal{Q}$ such that $\mathsf{Leak}(\mathcal{DS}_0, \mathcal{Q}) = \mathsf{Leak}(\mathcal{DS}_1, \mathcal{Q})$. As a response, the adversary $\mathcal{A}$ obtains $\mathsf{Enc}_K(\mathcal{DS}_b)$ and $\mathsf{Token}_K(\mathcal{Q})$.
3. **Output phase:** The adversary $\mathcal{A}$ outputs a value $\sigma \in \{0, 1\}$ which is defined as the output of the experiment.

4.2 aIND security

The adaptive indistinguishability-based security notion, first presented in [14] and known as aIND, considers a security experiment in which two entities interact with each other: a challenger and an adversary. The challenger first generates public and private parameters of the PSO scheme and sends its public parameters to the adversary. Upon receiving the public parameters, the adversary can gradually and adaptively issue challenge word pairs (w_0, w_1) for an arbitrary label l and construct datasets $\mathcal{DS}_0$ and $\mathcal{DS}_1$, and receive an updated version of the encrypted dataset $\widetilde{\mathcal{DS}}_b$ for a fixed bit b. The adversary can also issue adaptively the desired query and receive its token. Let $\mathcal{Q}$ be the list of set queries that the adversary has obtained their tokens at a given point of time. It should be noted that for the experiment to be meaningful, similar to the security notion naIND, it is essential that the adversary's choices are such that the relation $\mathsf{Leak}(\mathcal{DS}_0, \mathcal{Q}) = \mathsf{Leak}(\mathcal{DS}_1, \mathcal{Q})$, is maintained at every point in the experiment. An adversary who always maintains the mentioned relation is called a valid adversary. In such a setting, this security notion guarantees that a valid adversary is unable to distinguish between pairs $\big(\mathsf{Enc}_K(\mathcal{DS}_0), \mathsf{Token}_K(\mathcal{Q})\big)$ and $\big(\mathsf{Enc}_K(\mathcal{DS}_1), \mathsf{Token}_K(\mathcal{Q})\big)$ with a non-negligible advantage. In the following, the formal definition of this security notion is given.

Definition 3 (aIND security [14]) An PSO scheme such as $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Token}, \mathsf{Operate}, \mathsf{Retrieve})$ is called adaptively indistinguishable-based secure with respect to the leakage function Leak (abbreviated Leak-aIND-secure) if for every PPT valid adversary $\mathcal{A}$, there exists a negligible function ε such that

$$| \Pr[\mathsf{Exp}^{\mathsf{aIND}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1] - \Pr[\mathsf{Exp}^{\mathsf{aIND}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1] | \leq \varepsilon(\lambda),$$

where for each $b \in \{0,1\}$, the experiment $\mathsf{Exp}^{\mathsf{aIND}}_{\Pi,\mathcal{A}}(\lambda, b)$ is defined as follows.

1. **Setup phase:** The challenger Chal samples $(Param, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and sends the public parameters $Param$ to the adversary $\mathcal{A}$. In addition, two empty dataset $\mathcal{DS}_0, \mathcal{DS}_1$ and an empty list $\mathcal{Q}$ for the queries are initialized.
2. **Challenge phase:** The adversary $\mathcal{A}$ may adaptively issue $\mathsf{Enc}(\cdot,\cdot,\cdot)$ and $\mathsf{Token}(\cdot)$ queries, which are defined as follows:
   - $\mathsf{Enc}(w_0, , w_1, l)$: The pairs (w_0, l) and (w_1, l) are inserted into the dataset $\mathcal{DS}_0$ and $\mathcal{DS}_1$, respectively. Then, the challenger sends an encrypted-word $\mathsf{Enc}_K(w_b, l)$ to the adversary $\mathcal{A}$.
   - $\mathsf{Token}(q)$: The query q is inserted into the list $\mathcal{Q}$, and the challenger sends the set operation tokens $SOT \leftarrow \mathsf{Token}_K(q)$ to the adversary $\mathcal{A}$.
3. **Output phase:** The adversary $\mathcal{A}$ outputs a value $\sigma \in \{0,1\}$ which is defined as the output of the experiment.

4.3 Security relations

In this subsection, we study the implications and separations between the two security notions `naIND` and `aIND`. In particular, we show the relations "`naIND` $\not\Longrightarrow$ `aIND`" and "`aIND` $\Longrightarrow$ `naIND`", and conclude that the security notion `aIND` is strictly stronger than the security notion `naIND`. For this purpose, see Propositions 1 and 2.

Proposition 1 (`naIND` $\not\Longrightarrow$ `aIND`) *Let* Leak *be the leakage function defined in Section 3. Then, the* Leak-***naIND*** *security does not necessarily imply the* Leak-***aIND*** *security.*

Proof Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Token}, \mathsf{Operate}, \mathsf{Retrieve})$ be some Leak-`aIND`-secure PSO scheme. Based on this scheme, we construct an PSO scheme $\widetilde{\Pi} = (\widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Token}}, \widetilde{\mathsf{Operate}}, \widetilde{\mathsf{Retrieve}})$ which is yet Leak-`naIND`-secure, but it is not Leak-`aIND`-secure. The scheme $\widetilde{\Pi}$ is as follows:

- $\widetilde{\mathsf{KeyGen}}(1^\lambda)$**:** It runs $(Param, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and chooses a random word $w^* \in \{0,1\}^\lambda$, and outputs $(Param, \widetilde{K})$, where $\widetilde{K} = (K, w^*)$.
- $\widetilde{\mathsf{Enc}}_{\widetilde{K}}(l, w)$**:** It computes an encoded-word $\widetilde{w}$ as

$$\widetilde{w} = \widetilde{\mathsf{Enc}}_{\widetilde{K}}(w, l) = \begin{cases} (\mathsf{Enc}_K(w, l), w^*) & w \neq w^* \\ w^* & w = w^* \end{cases}.$$

It is easy to see that $\widetilde{\Pi}$ is still Leak-`naIND`-secure without any change in the leakage function, since the adversary's advantage only increases by $\frac{1}{2^\lambda}$ due to his ability to guess w^*. It is also easy to construct a Leak-`aIND`-attacker for $\widetilde{\Pi}$ with advantage one, since the adversary learns w^* after issuing the first query. He will then use it as one of his challenge words. □

Proposition 2 (`aIND` $\Longrightarrow$ `naIND`) *Let* Leak *be the leakage function defined in Section 3. If an PSO scheme is* Leak-***aIND****-secure, then it is* Leak-***naIND****-secure too.*

Proof The proof of this proposition is straightforward, as the IND-security experiment is essentially less restrictive than the naIND-security experiment. This proposition can also be proven using the proof by contradiction: given an adversary $\mathcal{A}$ to the naIND-security experiment, we can construct an adversary $\mathcal{B}$ to the aIND-security experiment (having the exact same advantage) as follows:

- At the beginning, the adversary $\mathcal{B}$ receives $\mathcal{DS}_0$, $\mathcal{DS}_1$ and a query list Q from the adversary $\mathcal{A}$,
- In the next step, based on the datasets $\mathcal{DS}_0$, $\mathcal{DS}_1$, the adaptive adversary issues the queries $\mathsf{Enc}(w_0, w_1, l)$, and constructs the encrypted dataset $\widetilde{\mathcal{DS}}$. Also, for every query $q \in Q$, he issues $\mathsf{Token}(q)$ and constructs a list of set operation tokens SOT^*.
- Finally, the adversary $\mathcal{B}$ sends the encrypted dataset $\widetilde{\mathcal{DS}}$ and the list of set operation tokens SOT^* to the adversary $\mathcal{A}$ related to the naIND-security experiment.

□

5 Simulation-based security notions

In this section, we first study the simulation-based security notions in both adaptive and non-adaptive versions. The non-adaptive version of this security notion was first presented in paper [13], and we review it for recall. However, we introduce its adaptive version for the first time, based on standard security notions in the literature. Finally, we investigate the relation between these two security notions and show that the adaptive version is strictly stronger than its non-adaptive version.

5.1 naSIM security

The non-adaptive simulation-based security notion, first presented in [13] and known as naSIM, considers an adversary $\mathcal{A}$, a simulator $\mathcal{S}$ and two experiments: *Real-world* and *Ideal-world*. In the real-world experiment, the adversary interacts with the PSO scheme in a non-adaptive approach: it first obtains the public parameters of the scheme and then, based on that, selects a dataset $\mathcal{DS}$ and a list of queries $\mathcal{Q}$ and sends them to the challenger. Finally, the challenger computes and returns the encrypted dataset $\widetilde{\mathcal{DS}}$ and set operation tokens SOT to the adversary. In the ideal-world experiment, the adversary, through the challenger, interacts non-adaptively with a simulator that executes PSO scheme. Similar to a real-world experiment, the adversary receives the public parameters and then based on them selects dataset $\mathcal{DS}$ and query list $\mathcal{Q}$ and sends them to the challenger. Next, the challenger computes the leakage profile $(\mathrm{Size}, \mathrm{SP}, \mathrm{OPP}, \mathrm{RP}, \mathrm{DP}) \leftarrow \mathsf{LeakP}(\mathcal{DS}, \mathcal{Q})$ and sends it to the simulator. In such a situation, the simulator must produce a view that is indistinguishable from the view produced in the real-world experiment. This indistinguishability guarantees that an adversary does not obtain information beyond the leakage included in the leakage profile. The formal definition of this security notion is given below.

Definition 4 (naSIM security [13]) An PSO scheme $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Token}, \mathsf{Operate}, \mathsf{Retrieve})$ is Leak-semantically-secure against non-adaptive adversaries if for every PPT adversary $\mathcal{A}$ there exist a PPT simulator $\mathcal{S}$ and a negligible function ε such that

$$| \Pr[\mathsf{Exp}^{\mathsf{Real}}_{\Pi,\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{Ideal}}_{\Pi,\mathcal{A},\mathcal{S}}(\lambda) = 1] | \leq \varepsilon(\lambda),$$

where the experiments $\mathsf{Exp}^{\mathsf{Real}}_{\Pi,\mathcal{A}}(\lambda)$ and $\mathsf{Exp}^{\mathsf{Ideal}}_{\Pi,\mathcal{A},\mathcal{S}}(\lambda)$ are defined as follows:

Real world ($\mathsf{Exp}^{\mathsf{Real}}_{\Pi,\mathcal{A}}(\lambda)$)**:**

1. **Setup phase:** The challenger samples $(Param, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$, and returns the public parameters $Param$ to the adversary $\mathcal{A}$.
2. **Challenge phase:** The adversary $\mathcal{A}$ takes the security parameter λ, chooses a dataset $\mathcal{DS}$ and a list of set queries $\mathcal{Q}$, and sends $\mathcal{DS}$ and $\mathcal{Q}$ to the challenger.
3. **Response phase:** The challenger computes $\widetilde{\mathcal{DS}} \leftarrow \mathsf{Enc}_K(\mathcal{DS})$ and $SOT \leftarrow \mathsf{Token}_K(\mathcal{Q})$, and returns them to the adversary $\mathcal{A}$.
4. **Output phase:** The adversary $\mathcal{A}$ outputs a value $\sigma \in \{0, 1\}$ which is defined as the output of the experiment.

Ideal world ($\mathsf{Exp}^{\mathsf{Ideal}}_{\Pi,\mathcal{A},\mathcal{S}}(\lambda)$)**:**

1. **Setup phase:** The simulator $\mathcal{S}$ produces the public parameters $Param^*$, which are given as input to the adversary $\mathcal{A}$.
2. **Challenge phase:** The adversary $\mathcal{A}$ takes the security parameter λ, chooses a dataset $\mathcal{DS}$ and a list of set queries $\mathcal{Q}$, and sends them to the challenger.
3. **Response phase:** The challenger computes the leakage profile $L \leftarrow \mathsf{Leak}(\mathcal{DS}, \mathcal{Q})$, and gives L as input to the simulator. The simulator $\mathcal{S}$ produces an encrypted dataset $\widetilde{\mathcal{DS}}^*$ and a list of set operation tokens SOT^*, which are then given to the adversary $\mathcal{A}$.
4. **Output phase:** The adversary $\mathcal{A}$ outputs a value $\sigma \in \{0, 1\}$ which is defined as the output of the experiment.

5.2 aSIM security

In this subsection, we introduce an adaptive version of the security notion naSIM. This security notion, similar to the non-adaptive variant, considers an adversary $\mathcal{A}$, a simulator $\mathcal{S}$ and two experiments: *real-world* and *ideal-world*. In the real-world experiment, the adversary $\mathcal{A}$ interacts with the PSO scheme and receives the public parameters. It can then adaptively issue a pair of word/set-label (w, l), and obtain an encrypted-word/set-label pair $(\tilde{l}, \tilde{w})$. The adversary can also adaptively issue a query q, and obtain a set operation token sot. In the ideal-world, however, the adversary $\mathcal{A}$ interacts with the simulator $\mathcal{S}$. It obtains the public parameters produced by $\mathcal{S}$. Then, the adversary can adaptively issue a pair of word/set-label (w, l), and obtain a pair of encrypted-word/set-label produced by the simulator using the leakage profile $(\text{Size}, \text{SP}, \text{OPP}, \text{RP}, \text{DP}) \leftarrow \mathsf{LeakP}(\mathcal{DS}, \mathcal{Q})$. The adversary can also adaptively issue a query q, and obtain a set operation token sot, that is produced by the simulator using the leakage profile $(\text{Size}, \text{SP}, \text{OPP}, \text{RP}, \text{DP}) \leftarrow \mathsf{LeakP}(\mathcal{DS}, \mathcal{Q})$. In such a situation, similar to the non-adaptive version, we expect the view produced by the simulator to be indistinguishable from the view produced in the real world. The formal definition of this security notion is as follows.

Definition 5 (aSIM security) An PSO scheme such as $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Token}, \mathsf{Operate}, \mathsf{Retrieve})$ is said to be adaptively simulation-based secure with respect to the leakage function Leak (abbreviated Leak-aSIM-secure) if for every PPT adversary $\mathcal{A}$ there exist a PPT simulator $\mathcal{S}$ and a negligible function ε such that

$$| \Pr[\mathsf{Exp}^{\mathsf{Real,aSIM}}_{\Pi,\mathcal{A}}(\lambda) = 1] - \Pr[\mathsf{Exp}^{\mathsf{Ideal,aSIM}}_{\Pi,\mathcal{A},\mathcal{S}}(\lambda) = 1] | \leq \varepsilon(\lambda),$$

where the experiments $\mathsf{Exp}^{\mathsf{Real,aSIM}}_{\Pi,\mathcal{A}}(\lambda)$ and $\mathsf{Exp}^{\mathsf{Ideal,aSIM}}_{\Pi,\mathcal{A},\mathcal{S}}(\lambda)$ are defined as follows.

Real-world $(\mathsf{Exp}^{\mathsf{Real,aSIM}}_{\Pi,\mathcal{A}}(\lambda))$**:**

1. **Setup phase:** The challenger samples $(Param, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$, and returns the public parameters $Param$ to the adversary $\mathcal{A}$.
2. **Challenge phase:** The adversary $\mathcal{A}$ may adaptively issue $\mathsf{Enc}(\cdot,\cdot)$ and $\mathsf{Token}(\cdot)$ queries and receive the output.
3. **Output phase:** The adversary $\mathcal{A}$ outputs a value $\sigma \in \{0,1\}$ which is defined as the output of the experiment.

Ideal-world $(\mathsf{Exp}^{\mathsf{Ideal,aSIM}}_{\Pi,\mathcal{A},\mathcal{S}}(\lambda))$**:**

1. **Setup phase:** The simulator $\mathcal{S}$ produces the public parameters $Param^*$, which are given as input to the adversary $\mathcal{A}$. An empty dataset $\mathcal{DS}$ and an empty list $\mathcal{Q}$ of set labels are initialized.
2. **Challenge phase:** The adversary $\mathcal{A}$ may adaptively issue $\mathsf{Enc}(\cdot,\cdot)$ and $\mathsf{Token}(\cdot)$ queries, which are defined as follows:
   - $\mathsf{Enc}(w,l)$: The pair (w,l) is inserted into the dataset $\mathcal{DS}$, and the simulator $\mathcal{S}$ obtains the leakage profile $L \leftarrow \mathsf{Leak}(\mathcal{DS},\mathcal{Q})$. Finally, the simulator $\mathcal{S}$ sends an encrypted-word $\widetilde{w}^*$ to the adversary $\mathcal{A}$.
   - $\mathsf{Token}(q)$: The query $q = (op, l_i, l_j)$ is inserted into the list $\mathcal{Q}$, and the simulator $\mathcal{S}$ obtains the leakage profile $L \leftarrow \mathsf{Leak}(\mathcal{DS},\mathcal{Q})$. Finally, the simulator $\mathcal{S}$ sends a list of set operation tokens SOT^* to the adversary $\mathcal{A}$.
3. **Output phase:** The adversary $\mathcal{A}$ outputs a value $\sigma \in \{0,1\}$ which is defined as the output of the experiment.

5.3 Security relations

In this subsection, we study the implications and separations between the security notions naSIM and aSIM. In particular, we show the relations "naSIM $\not\Longrightarrow$ aSIM" and "aSIM $\Longrightarrow$ naSIM", and conclude that the security notion aSIM is strictly stronger than the security notion naSIM. For this purpose, see Propositions 3 and 4.

Proposition 3 (naSIM $\not\Longrightarrow$ aSIM) *Let* Leak *be the leakage function defined in Section 3. Then, the* Leak-***naSIM*** *security does not necessarily imply the* Leak-***aSIM*** *security.*

Proof The proof of this proposition is similar to Proposition 1 for separating concepts naIND and aIND. Let $\Pi = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Token}, \mathsf{Operate}, \mathsf{Retrieve})$ be some Leak-aSIM-secure PSO scheme. Based on this scheme, we construct an PSO scheme $\widetilde{\Pi} = (\widetilde{\mathsf{KeyGen}}, \widetilde{\mathsf{Enc}},$

$\widetilde{\mathsf{Token}}, \widetilde{\mathsf{Operate}}, \widetilde{\mathsf{Retrieve}}$) which is yet Leak-naSIM-secure, but it is not Leak-aSIM-secure. The scheme $\widetilde{\Pi}$ is as follows:

- $\widetilde{\mathsf{KeyGen}}(1^\lambda)$**:** It runs $(Param, K) \leftarrow \mathsf{KeyGen}(1^\lambda)$ and chooses a random word $w^* \in \{0,1\}^\lambda$, and outputs $(Param, \widetilde{K})$, where $\widetilde{K} = (K, w^*)$.
- $\widetilde{\mathsf{Enc}}_{\widetilde{K}}(l, w)$**:** It computes an encoded-word $\widetilde{w}$ as

$$\widetilde{w} = \widetilde{\mathsf{Enc}}_{\widetilde{K}}(w, l) = \begin{cases} (\mathsf{Enc}_K(w, l), w^*) & w \neq w^* \\ w^* & w = w^* \end{cases}.$$

It is easy to see that $\widetilde{\Pi}$ is still naSIM-secure without any change in the leakage function, since the adversary's advantage only increases by $\frac{1}{2^\lambda}$ due to his ability to guess w^*. It is also easy to construct a aSIM-attacker for $\widetilde{\Pi}$ with advantage one, since the adversary learns w^* after issuing the first query. He will then use it as one of his challenge words. □

Proposition 4 (aSIM $\Longrightarrow$ naSIM) *Let* Leak *be the leakage function defined in Section 3. If an PSO scheme is* Leak-***aSIM****-secure, then it is* Leak-***naSIM****-secure too.*

Proof The proof of this proposition is straightforward, as the aSIM-security experiment is essentially less restrictive than the naSIM-security experiment. This proposition can also be proven using the proof by contradiction: given an adversary $\mathcal{A}$ to the naSIM-security experiment, we can construct an adversary $\mathcal{B}$ to the aSIM-security experiment (having the exact same advantage) as follows:

- At the beginning, the adversary $\mathcal{B}$ receives a dataset $\mathcal{DS}$ and a query list Q from the adversary $\mathcal{A}$,
- In the next step, based on the dataset $\mathcal{DS}$, the adaptive adversary $\mathcal{B}$ issues the queries $\mathsf{Enc}(w_0, w_1, l)$, and constructs the encrypted dataset $\widetilde{\mathcal{DS}}$. Also, for every query $q \in Q$, he issues $\mathsf{Token}(q)$ and constructs a list of set operation tokens SOT^*.
- Finally, the adversary $\mathcal{B}$ sends the encrypted dataset $\widetilde{\mathcal{DS}}$ and the list of set operation tokens SOT^* to the adversary $\mathcal{A}$ related to the naSIM-security experiment.

□

6 Security relations between IND and SIM

In this section, we study the relations between the indistinguishability-based and simulation-based security notions. More precisely, in Propositions 5 and 6 we show the relations "naIND $\Longrightarrow$ naSIM" and "naSIM $\Longrightarrow$ naIND", and consequently these two notions are equivalent to each other. We also show in Proposition 7 that security notion aSIM implies security notion aIND. However, the relation "aIND $\overset{?}{\Longrightarrow}$ aSIM" remains an open problem.

To prove relation "naIND $\Longrightarrow$ naSIM" (Proposition 5), we need two concepts: History and Non-singular history, which were introduced for the first time in [14]. For this purpose, we first review the formal definitions of these two concepts.

Definition 6 (History [14]) A tuple $H = (\mathcal{DS}, Q)$ is called a l-query history, where $\mathcal{DS}$ represents a dataset and Q represents a vector of l set operations ($Q = (q_1, \cdots, q_l)$).

Definition 7 (Non-singular history [14]) We say that a history H is non-singular if:

- there exists at least one history $H' \neq H$ such that $\mathsf{Leak}(H) = \mathsf{Leak}(H')$,
- such a history can be found in polynomial-time given $\mathsf{Leak}(H)$.

Proposition 5 ($\texttt{naIND} \implies \texttt{naSIM}$) *Let* Leak *be the leakage function defined in Section 3. If an PSO scheme is* Leak-***naIND****-secure, then it is* Leak-***naSIM****-secure too.*

Proof We show that if there exists a PPT adversary $\mathcal{A}$ such that for every PPT simulator $\mathcal{S}$, it is able to distinguish between the outputs $\mathsf{Exp}^{\mathsf{Real,aSIM}}_{\Pi,\mathcal{A}}(\lambda)$ and $\mathsf{Exp}^{\mathsf{Ideal,aSIM}}_{\Pi,\mathcal{A},\mathcal{S}}(\lambda)$, then there exists a PPT adversary $\mathcal{B}$ that succeeds in the experiment $\mathsf{Exp}^{\mathtt{naIND}}_{\Pi,\mathcal{A}}(\lambda, b)$ with non-negligible probability over $\frac{1}{2}$.

Let the non-singular history $H_0 = (\mathcal{DS}, Q)$ be the output of $\mathcal{A}(\lambda)$. Therefore, there exists at least one history $H_1 \neq H_0$ such that $\mathsf{Leak}(H_0) = \mathsf{Leak}(H_1)$, and also it can be found efficiently. Without loss of generality, we assume that the adversary $\mathcal{A}$ outputs 0 when given the output of the experiment $\mathsf{Exp}^{\mathsf{Real,aSIM}}_{\Pi,\mathcal{A}}(\lambda)$.

The adversary $\mathcal{B}$ acts as follows:

- In the first step, the adversary $\mathcal{B}$ computes H_1 using $\mathsf{Leak}(H_0)$,
- In the second step, the adversary $\mathcal{B}$ gives (H_0, H_1) to the adversary $\mathcal{A}$.
- It remains to analyze $\mathcal{B}$'s success probability. Since b is chosen uniformly at random, we have:

$$\Pr[\mathsf{Exp}^{\mathtt{naIND}}_{\Pi,\mathcal{A}}(\lambda, b) = 0] = \frac{1}{2}\Big(\Pr[\mathsf{Exp}^{\mathtt{naIND}}_{\Pi,\mathcal{A}}(\lambda, b) = 0 \mid b = 0] + \Pr[\mathsf{Exp}^{\mathtt{naIND}}_{\Pi,\mathcal{A}}(\lambda, b) = 0 \mid b = 1]\Big)$$

If $b = 0$ occurs then adversary $\mathcal{B}$ succeeds if and only if the adversary $\mathcal{A}$ outputs 0. Therefore, we have:

$$\Pr[\mathsf{Exp}^{\mathsf{aIND}}_{\Pi,\mathcal{A}}(\lambda, b) = 0 \mid b = 0] = \Pr[\mathsf{Exp}^{\mathsf{Real,aSIM}}_{\Pi,\mathcal{A}}(\lambda) = 0]$$

On the other hand, if $b = 1$ then $\mathcal{B}$ succeeds if and only if the adversary $\mathcal{A}$ outputs 1. In this case, we have:

$$\Pr[\mathsf{Exp}^{\mathsf{aIND}}_{\Pi,\mathcal{A}}(\lambda, b) = 0 \mid b = 1] = 1 - \Pr[\mathsf{Exp}^{\mathsf{Ideal,aSIM}}_{\Pi,\mathcal{A}}(\lambda) = 1]$$

By combining above equations, we have:

$$\begin{aligned}\Pr[\mathsf{Exp}^{\mathsf{aIND}}_{\Pi,\mathcal{A}}(\lambda, b) = 0] &= \frac{1}{2}\Big(\Pr[\mathsf{Exp}^{\mathsf{Real,aSIM}}_{\Pi,\mathcal{A}}(\lambda) = 0] + 1 - \Pr[\mathsf{Exp}^{\mathsf{Ideal,aSIM}}_{\Pi,\mathcal{A}}(\lambda) = 0]\Big) \\ &= \frac{1}{2} + \frac{1}{2}\Big(\,|\,\Pr[\mathsf{Exp}^{\mathsf{Real,aSIM}}_{\Pi,\mathcal{A}}(\lambda) = 0] - \Pr[\mathsf{Exp}^{\mathsf{Ideal,aSIM}}_{\Pi,\mathcal{A}}(\lambda) = 0]\,|\,\Big) \geq \frac{1}{2} + \mu(\lambda)\end{aligned}$$

where $\mu(\lambda)$ is a non-negligible function in λ, and the inequality follows from our assumption about the adversary $\mathcal{A}$.

□

Proposition 6 (naSIM $\implies$ naIND) *Let* Leak *be the leakage function defined in Section 3. If an PSO scheme is* Leak-*naSIM*-*secure, then it is* Leak-*naIND*-*secure too.*

Proof Let $\mathcal{A}$ be an adversary for the naIND security experiment, we can construct an adversary $\mathcal{B}$ for the naSIM experiment as follows:

- The adversary $\mathcal{B}$ gives the security parameter λ to the adversary $\mathcal{A}$, and receives two datasets $\mathcal{DS}_0, \mathcal{DS}_1$, and a list of queries $\mathcal{Q}$ such that $\mathsf{Leak}(\mathcal{DS}_0, \mathcal{Q}) = \mathsf{Leak}(\mathcal{DS}_1, \mathcal{Q})$.
- The adversary $\mathcal{B}$ samples $c \leftarrow \{0, 1\}$ and outputs $(\mathcal{DS}_c, \mathcal{Q})$,
- Finally, the adversary $\mathcal{B}$ receives the output of the naIND security experiment (i.e., σ), and outputs $\sigma \oplus c$.

Therefore, for any simulator $\mathcal{S}$ we have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{naIND}}_{\Pi,\mathcal{A}}(\lambda) =& \mid \Pr[\mathsf{Exp}^{\mathsf{naIND}}_{\Pi,\mathcal{A}}(\lambda, 0) = 1] - \Pr[\mathsf{Exp}^{\mathsf{naIND}}_{\Pi,\mathcal{A}}(\lambda, 1) = 1] \mid \\ =& \mid \Pr[\mathsf{Exp}^{\mathsf{Real},\mathsf{naSIM}}_{\Pi,\mathcal{B}}(\lambda) = 1 \mid c = 0] - \\ & \Pr[\mathsf{Exp}^{\mathsf{Real},\mathsf{naSIM}}_{\Pi,\mathcal{B}}(\lambda) = 1 \mid c = 1] \mid .\end{aligned}$$

Also, since $\mathsf{Leak}(\mathcal{DS}_0, \mathcal{Q}) = \mathsf{Leak}(\mathcal{DS}_1, \mathcal{Q})$, we have

$$\Pr[\mathsf{Exp}^{\mathsf{Ideal},\mathsf{naSIM}}_{\Pi,\mathcal{S}}(\lambda) = 1 \mid c = 0] = \Pr[\mathsf{Exp}^{\mathsf{Ideal},\mathsf{naSIM}}_{\Pi,\mathcal{S}}(\lambda) = 1 \mid c = 1],$$

and hence for $\mathsf{Adv}^{\mathsf{naIND}}_{\Pi,\mathcal{A}}(\lambda)$ we have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{naIND}}_{\Pi,\mathcal{A}}(\lambda) =& \mid \Pr[\mathsf{Exp}^{\mathsf{Real},\mathsf{naSIM}}_{\Pi,\mathcal{B}}(\lambda) = 1 \mid c = 0] - \\ & \Pr[\mathsf{Exp}^{\mathsf{Ideal},\mathsf{naSIM}}_{\Pi,\mathcal{S}}(\lambda) = 1 \mid c = 0] + \\ & \Pr[\mathsf{Exp}^{\mathsf{Ideal},\mathsf{naSIM}}_{\Pi,\mathcal{S}}(\lambda) = 1 \mid c = 1] - \\ & \Pr[\mathsf{Exp}^{\mathsf{Real},\mathsf{naSIM}}_{\Pi,\mathcal{B}}(\lambda) = 1 \mid c = 1] \mid .\end{aligned}$$

Finally, since the adversary $\mathcal{B}$ inverts the output of adversary $\mathcal{A}$ in case that $c = 1$, we have

$$\begin{aligned}\mathsf{Adv}^{\mathsf{naIND}}_{\Pi,\mathcal{A}}(\lambda) =& \mid \Pr[\mathsf{Exp}^{\mathsf{Real},\mathsf{naSIM}}_{\Pi,\mathcal{B},\mathcal{S}}(\lambda) = 1 \mid c = 0] \; - \\ & \Pr[\mathsf{Exp}^{\mathsf{Ideal},\mathsf{naSIM}}_{\Pi,\mathcal{B},\mathcal{S}}(\lambda) = 1 \mid c = 0] \; + \\ & \Pr[\mathsf{Exp}^{\mathsf{Real},\mathsf{naSIM}}_{\Pi,\mathcal{B},\mathcal{S}}(\lambda) = 1 \mid c = 1] \mid - \\ & \Pr[\mathsf{Exp}^{\mathsf{Ideal},\mathsf{naSIM}}_{\Pi,\mathcal{B},\mathcal{S}}(\lambda) = 1 \mid c = 1] \\ =& \; 2 \cdot \mathsf{Adv}^{\mathsf{naSIM}}_{\Pi,\mathcal{B},\mathcal{S}}(\lambda).\end{aligned}$$

Using the assumption that the scheme Π is Leak-naSIM-secure, there exists a simulator $\mathcal{S}$ such that $\mathsf{Adv}^{\mathsf{naSIM}}_{\Pi,\mathcal{B},\mathcal{S}}(\lambda)$ is negligible. Therefore, $\mathsf{Adv}^{\mathsf{naIND}}_{\Pi,\mathcal{A}}(\lambda)$ is negligible, and the proof is complete. □

Proposition 7 (aSIM $\implies$ aIND) *Let* Leak *be the leakage function defined in Section 3. If an PSO scheme is* Leak-*aSIM*-*secure, then it is also* Leak-*aIND*-*secure.*

Proof The proof idea is similar to non-adaptive variant (i.e., Proposition 6). In a general and intuitive way, the idea is that the adversary cannot distinguish between the encrypted dataset $\mathcal{DS}_0$ and the encrypted dataset generated by the simulator, and between the encrypted dataset generated by the simulator and the encrypted dataset $\mathcal{DS}_1$. Similar to Proposition 6, let $\mathcal{A}$ be an adversary for the aIND security experiment, we can construct an adversary $\mathcal{B}$ for aSIM experiment as follows:

- The adversary $\mathcal{B}$ samples $c \leftarrow \{0,1\}$,
- The adversary $\mathcal{B}$ converts each query of the form $\mathsf{Enc}(w_0, w_1, l)$ into a query of the form $\mathsf{Enc}_K(w_c, l)$,
- Finally, when the adversary $\mathcal{A}$ halts and outputs σ, then the adversary $\mathcal{B}$ outputs $\sigma \oplus c$.

Using a similar argument to Proposition 6, we can show that $\mathsf{Adv}^{\mathtt{aSIM}}_{\Pi,\mathcal{B},\mathcal{S}}(\lambda)$ is negligible. □

7 The available PSO constructions

In this section, for comprehensiveness and completeness of the topics studied in this paper, we provide a detailed overview of the available PSO schemes.

Table 1 shows the available PSO schemes in terms of the cryptographic tool used, the underlying hardness assumption, the time complexity of the set operation algorithm (for m set of length n in terms of group operations), and the achieved security level. As it can be seen in Table 1, all available PSO constructions are pairing-based. These tools are known as blocks with moderate overhead compared to other possibilities, and therefore it is important to provide constructions with tools that impose lower overhead. In addition, Table 1 also shows that no PSO construction has yet been introduced for security level aSIM, and its study in future research could be important.

Table 1: A detailed overview of the available PSO schemes.

Scheme	Tools	Assumption	Operation time	Security			
				naIND	naSIM	aIND	aSIM
Rafiee and Khazaei [13] (PSOI)	Pairing	matix-DDH [16]	$(4n)$ BM	✓	✓	–	–
Rafiee and Khazaei [13] (PSOII)	Pairing	XDHV [13]	$(2n)$ BM	✓	✓	–	–
Rafiee [14]	Pairing	DDH1 [17]	$(2n)$ BM	✓	✓	✓	–

n: the maximum size of each set in the dataset, BM: bilinear map evaluation.

8 Conclusion and future works

In this paper, we reviewed the syntax and security notions of PSO schemes as a cryptographic primitive that enables a user to securely outsource his dataset to a cloud server and then securely execute his set queries on it. We then introduced two new natural security notions for these schemes: naIND and aSIM. We also studied and proved the implications and separations between all these security notions. In addition to these, we have listed all the available PSO schemes along with their security level.

The introduction of efficient PSO constructions and specifically construction with the security level aSIM, which has not been proposed so far, can be considered as future research directions. Also, the security relation $\texttt{aIND} \overset{?}{\Longrightarrow} \texttt{aSIM}$ remains as an open problem and the research on it can complete the results obtained in this paper.

Funding
The author declares that he did not receive any funds, grants, or other support during the preparation of this manuscript.

Availability of data and materials
Not applicable.

Declarations

Ethics approval and consent to participate
This article does not contain any studies with human participants or animals performed by any of the authors.

Competing interests
The authors declare no competing interests.

References

[1] Tabrizchi, H., Kuchaki Rafsanjani, M.: A survey on security challenges in cloud computing: issues, threats, and solutions. The journal of supercomputing **76**(12), 9493–9532 (2020)

[2] Qiang, W.: Performance and security in cloud computing. The Journal of Supercomputing **75**(1), 1–3 (2019)

[3] Ghorbel, A., Ghorbel, M., Jmaiel, M.: Privacy in cloud computing environments: a survey and research challenges. The Journal of Supercomputing **73**(6), 2763–2800 (2017)

[4] Huang, B., Gao, J., Li, X.: Efficient lattice-based revocable attribute-based encryption against decryption key exposure for cloud file sharing. Journal of Cloud Computing **12**(1), 37 (2023)

[5] Sun, Y., Han, L., Bi, J., Tan, X., Xie, Q.: Verifiable attribute-based keyword search scheme over encrypted data for personal health records in cloud. Journal of Cloud Computing **12**(1), 77 (2023)

[6] Yang, N., Zhou, Q., Huang, Q., Tang, C.: Multi-recipient encryption with keyword search without pairing for cloud storage. Journal of Cloud Computing **11**(1), 10 (2022)

[7] Wu, J., Zhang, K., Wei, L., Gong, J., Ning, J.: Practical searchable symmetric encryption for arbitrary boolean query-join in cloud storage. IEEE Transactions on Information Forensics and Security (2024)

[8] Deng, Q., Chen, L., Zhu, Y., Mu, Y.: Leakage reduced searchable symmetric encryption for multi-keyword queries. IEEE Transactions on Cloud Computing (2025)

[9] Du, K., Wang, J., Wu, J., Wang, Y.: Scalable equi-join queries over encrypted database. In: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, pp. 4002–4016 (2024)

[10] Rafiee, M.: Multi-adjustable join schemes with adaptive indistinguishably security. IEEE Transactions on Dependable and Secure Computing **21**(4), 4024–4034 (2023)

[11] Khazaei, S., Rafiee, M.: Towards more secure constructions of adjustable join schemes. IEEE Transactions on Dependable and Secure Computing **19**(2), 1078–1089 (2020)

[12] Mironov, I., Segev, G., Shahaf, I.: Strengthening the security of encrypted databases: Non-transitive joins. In: Theory of Cryptography Conference, pp. 631–661 (2017). Springer

[13] Rafiee, M., Khazaei, S.: Private set operations over encrypted cloud dataset and applications. The Computer Journal **64**(8), 1145–1162 (2021)

[14] Rafiee, M.: Towards more secure constructions of private set operation schemes. Cryptology ePrint Archive, Paper 2025/1360 (2025). https://eprint.iacr.org/2025/1360

[15] Rafiee, M., Khazaei, S.: Security of multi-adjustable join schemes: Separations and implications. IEEE Transactions on Dependable and Secure Computing **19**(4), 2535–2545 (2021)

[16] Escala, A., Herold, G., Kiltz, E., Ràfols, C., Villar, J.: An algebraic framework for diffie–hellman assumptions. Journal of cryptology **30**(1), 242–288 (2017)

[17] Scott, M.: Authenticated id-based key exchange and remote log-in with simple token and pin number. Cryptology ePrint Archive (2002)