Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling (ICAPS 2017)

# The Limits of Strong Privacy Preserving Multi-Agent Planning

**Jan Tožička, Michal Štolba, Antonín Komenda**
{tozicka,stolba,komenda}@agents.fel.cvut.cz
Department of Computer Science, Faculty of Electrical Engineering,
Czech Technical University in Prague, Czech Republic

## Abstract

Multi-agent planning using MA-STRIPS-related models is often motivated by the preservation of private information. Such motivation is not only natural for multi-agent systems, but it is one of the main reasons, why multi-agent planning (MAP) problems cannot be solved centrally. In this paper, we analyze privacy-preserving multi-agent planning (PP-MAP) from the perspective of secure multiparty computation (MPC). We discuss the concept of strong privacy and its implications and present two variants of a novel planner, provably strong privacy-preserving in general. As the main contribution, we formulate the limits of strong privacy-preserving planning in the terms of privacy, completeness and efficiency and show that, for a wide class of planning algorithms, all three properties are not achievable at once. Moreover, we provide a restricted variant of strong privacy based on equivalence classes of planning problems and show that an efficient, complete and strong privacy-preserving planner exists for such restriction.

## Introduction

Classical planning is able to find plans for a single agent, but also for multiple agents, vehicles, robots, etc. The difference between classical planning for multiple agents and privacy-preserving multi-agent planning is not only the physical distribution of the planning process, but most crucially the preservation of private information of the planning agents. Consider for example a consortium of businesses, which need to coordinate their actions in a joint project, but which do not want to disclose all their inner information and processes. Another example is a company working together with a hospital which cannot disclose the data of its patients, but wants to utilize the company's know-how. Last, but not least, consider a military coalition operation, which needs to be coordinated without disclosing classified information.

Although privacy is an important aspect of multi-agent planning it is often neglected in the literature. A rigorous definition of privacy for PP-MAP was proposed in (Nissim and Brafman 2014; Brafman 2015). Recently, there have been a number of works aiming at PP-MAP. First of all, a more secure version of the MAFS algorithm, Secure-MAFS (Brafman 2015) and its implementation and generalization Macro-MAFS (Maliah, Shani, and Brafman 2016). The two algorithms aim to communicate less states and thus also reveal less information by learning the possible public transitions of other agents. A recent work introduces the concept of information leakage based on transition systems and also present a theoretical class of privacy preserving algorithms (Stolba, Tozicka, and Komenda 2016a; Štolba, Tožička, and Komenda 2016b).

Somewhat orthogonal are the works introducing novel concepts of privacy, such as agent privacy (Maliah, Brafman, and Shani 2016), where the number and identity of agents is hidden, and object cardinality privacy (Maliah, Shani, and Stern 2016), where the types of objects may be revealed, but not the numbers of instances of each type. We do not include these concepts in our current analysis.

Along the direction of the research of Secure-MAFS, we focus on proposing a complete planner provably strong privacy-preserving in any multi-agent planning problem compatible with the MA-STRIPS model. Although we show that such planner exists, we also show that only for the price of either inefficiency, rendering it practically unusable, or incompleteness which may be the only practical approach to strong-privacy preserving planning. This result is a formal reformulation and generalization of the statement by (Brafman 2015):

> "A search-based approach in which intermediate search nodes are shared among agents is unlikely to be strongly private always."

In this work, we prove the theoretical bounds of strong-privacy preserving planning based on the most common MAP paradigms, such as distributed state-space search. We also propose a finer-grained definition of strong privacy, where some aspects of the problem are known a priori (e.g., that it is a logistics task) and only the details are left strongly private. We show that a complete, efficient and strong privacy-preserving planner exists for such restricted notion of strong privacy.

## Multi-Agent Planning (MAP)

Let us define the MAP problem based on the MA-STRIPS (Brafman and Domshlak 2008) formalism. For a set of $n$ agents $\mathcal{A}$, a MAP problem $\mathcal{M} = \{\Pi_i\}_{i=1}^n$ is a set of agents'

local STRIPS problems. An agent problem of agent $i$ is defined as

$$\Pi_i = \left\langle P_i = P^{\mathsf{pub}} \cup P_i^{\mathsf{priv}}, A_i, s_I \cap P_i, s_\star \cap P_i \right\rangle,$$

where $P_i \subseteq P$ is a set of propositions (facts) partitioned into disjunctive sets $P^{\mathsf{pub}}$ and $P_i^{\mathsf{priv}}$ of public (common to all agents) and private (of agent $i$) facts respectively. The state $s_I \subseteq P$ is the initial state and $s_\star \subseteq P^{\mathsf{pub}}$ represents the goal condition (that is, all facts in $s_\star$ must hold in any goal state).

An action $a \in A_i$ is defined using the classical STRIPS syntax and semantics

$$a = \langle \mathsf{pre}(a), \mathsf{add}(a), \mathsf{del}(a), \mathsf{lbl}(a) \rangle,$$

where $\mathsf{pre}(a) \subseteq P_i$, $\mathsf{add}(a) \subseteq P_i$ and $\mathsf{del}(a) \subseteq P_i$ are the sets of preconditions, add effects and delete effects respectively and $\mathsf{lbl}(a)$ is a unique action identifier. An action $a \in A_i$ is public if either $\mathsf{pre}(a) \cap P^{\mathsf{pub}} \neq \emptyset$, $\mathsf{add}(a) \cap P^{\mathsf{pub}} \neq \emptyset$ or $\mathsf{del}(a) \cap P^{\mathsf{pub}} \neq \emptyset$, otherwise $a$ is private.

The set $A_i$ of actions comprises of three pairwise disjoint sets: a set $A_i^{\mathsf{priv}}$ of private actions of agent $i$, a set $A_i^{\mathsf{pub}}$ of public actions of agent $i$ and a set $A_i^{\mathsf{proj}}$ of public projections of public actions of other agents. Note that each private and public action belongs to exactly one agent, formally $A_i^{\mathsf{priv}} \cap A_j^{\mathsf{priv}} = \emptyset$ and $A_i^{\mathsf{pub}} \cap A_j^{\mathsf{pub}} = \emptyset$ for all $i \neq j$. A public projection $a^\triangleright$ of an action $a \in A_i^{\mathsf{pub}}$ is $a$ with precondition and effect restricted only to the facts in $P^{\mathsf{pub}}$ and without the label, that is

$$a^\triangleright = \left\langle \mathsf{pre}(a) \cap P^{\mathsf{pub}}, \mathsf{add}(a) \cap P^{\mathsf{pub}}, \mathsf{del}(a) \cap P^{\mathsf{pub}} \right\rangle.$$

The projected actions are shared among all agents except for the owner of the original action. A public projection of $\mathcal{M}$ is a STRIPS problem

$$\mathcal{M}^\triangleright = \left\langle P^{\mathsf{pub}}, A_i^\triangleright, s_I^\triangleright, s_\star^\triangleright \right\rangle,$$

where

$$A_i^\triangleright = A_i \cup \{a^\triangleright | a \in \bigcup_{j=1}^{n} A_j^{\mathsf{pub}} \, s.t. \, j \neq i\}.$$

*Local plan* $\pi_i$ is a solution to $\Pi_i$ and a *global plan* $\{\pi_i\}_{i=1}^n$ is a solution to the MAP problem $\mathcal{M}$. A *public plan* $\pi_i^\triangleright$ is a projection of $\pi_i$ such that all public actions $a \in \pi_i \cap A_i^{\mathsf{pub}}$ are replaced by their public projections $a^\triangleright$ and all private actions $a' \in \pi_i \cap A_i^{\mathsf{priv}}$ are removed.

A public plan $\pi_j^\triangleright$ is $i$-extensible, if by adding private actions from $A_i^{\mathsf{priv}}$ to $\pi_j^\triangleright$ and replacing all projections $a^\triangleright$ s.t. $a \in A_i^{\mathsf{pub}}$ by $a$, we obtain a local solution to $\Pi_i$. According to (Tozicka et al. 2016), a public plan which is $i$-extensible by all agents $i \in 1, ..., n$ is a global solution to $\mathcal{M}$ (can be extended by all agents to form $\{\pi_i\}_{i=1}^n$). Note that for a global plan $\{\pi_i\}_{i=1}^n$, $\pi_i^\triangleright = \pi_j^\triangleright$ holds for all $i, j$, that is, the global plan is coordinated.

Based on (Tozicka et al. 2016), a set $S$ of plans for a (STRIPS) planning problem $\Pi$ (that is, $\Pi_i$ or $\mathcal{M}^\triangleright$) can be represented by a planning state machine (PSM).

**Definition 1. (PSM)** Let $\Pi = \langle P, A, s_I, s_\star \rangle$ be a STRIPS planning problem and $S$ a set of solutions of $\Pi$. A planning state machine (PSM) $\Gamma(S) = \langle \Sigma, N, s_I, \delta, F \rangle$ is a deterministic finite automaton (DFA) where the alphabet $\Sigma$ contains the STRIPS labels of the actions in $a \in A$ s.t. $\Sigma = \{\mathsf{lbl}(a) : a \in A\}$, states are sets of facts $N \subseteq 2^P$ with $s_I \in N$, transitions satisfy that $\delta(s, \mathsf{lbl}(a)) = s'$ iff the action $a$ transforms the state $s$ into another state $s'$ and accepting states are $F = \{s \in N : s_\star \subseteq s\}$. The PSM $\Gamma(S)$ accepts a sequence of actions $\pi$ iff $\pi \in S$.

A plan $\pi$ is accepted by $\Gamma(S)$ if it is a solution to $\Pi$ and $\pi \in S$. If $\Gamma(S)$ contains all solution to $\Pi$ we call it a full PSM and denote it $\Gamma(\Pi)$. An important advantage of the PSM structure over a set of plans is that a PSM remains finite even if $S$ is infinite and it is also possible to construct it in finite time, even though this time can be exponential in the size of the problem.

A public projection of PSM $\Gamma(S)$ is $\Gamma(S)^\triangleright$, where each state $s$ is replaced with a public projection $s^\triangleright$ and each transition representing a private action is replaced with an $\epsilon$-transition. The $\epsilon$-transitions are then eliminated using standard DFA algorithm and thus the PSM is minimized.

## Secure Multiparty Computation (MPC)

Secure multiparty computation (MPC) (Yao 1982) is a subfield of cryptography, which studies computing a function $f$ by a set of $n$ parties $p_1, ..., p_n$ such that each $p_i$ knows part of the input of $f$. The goal of MPC is to compute $f$ in such a way that no party $p_i$ learns more information about the inputs of other parties than what can be learned from the output of $f$. Clearly, PP-MAP is an instance of MPC, where $\Pi_i$ are the respective inputs and the global plan is the desired output.

In MPC, assumptions are typically placed on the participating parties (agents in our case) and their communication and computation capabilities. We assume that there is no trusted third-party and that the planning agents are semi-honest (or honest but curious). This means, as opposed to malicious agents that every agent follows the rules of the computation protocol based on its input data, but after the computation is finished, it is allowed to use any information it has received during the protocol to compromise the privacy. Regarding the communication model, we assume asynchronous communication, where all messages between each two agents retain the order in which they were sent.

The computation power of the agents (which can be used to infer additional knowledge from the executed protocol) is typically seen either as unbounded, in which case we are talking about *information-theoretic security*, or polynomial-time bounded, which is the case of *computational security*. When applied to PP-MAP, the notion of polynomial-time bounded adversary is somewhat less suitable, as the planning itself is not polynomial (but PSPACE-complete). Nevertheless, for planning problems which can be practically solved, we can keep the cryptographic assumptions (such as that the factoring of large integers is hard), for which the polynomial-time bound is typically used.

There are basically two approaches to multi-agent

planning based on the MPC techniques. The first approach is to encode planning in some general MPC technique such as cryptographic circuits (Yao 1986), oblivious RAM (ORAM) (Goldreich 1987) or blind Turing machine (BlindTM) (Rass, Schartner, and Brodbeck 2015). According to (Boyle and Naor 2016), ORAM can be used to encode MPC with at least superlogarithmic overhead. Nevertheless, it is not clear how exactly PP-MAP would be encoded and what the actual overhead would be. Similarly, it is not clear how and with what overhead a BlindTM would encode PP-MAP. For now, we leave this direction of research for future work. The second approach is to devise a specific PP-MAP algorithm based on MPC primitives. There is a number of solutions for a related problem, shortest path in a graph, e.g., (Brickell and Shmatikov 2005), which all have one major flaw. They solve the shortest path problem for an explicit graph (typically represented by a matrix), which is not feasible for larger planning problems.

In this paper, we follow the second direction and propose a PP-MAP algorithm (or rather a family of algorithms) based on private set intersection (PSI), which is a well known primitive in MPC. Several approaches to computationally secure PSI has been proposed in (Pinkas et al. 2015; Jarecki and Liu 2010). An information-theoretic approach was proposed in (Li and Wu 2007) which provides unconditional security, as long as at least $n/2$ parties are semi-honest. Another MPC primitive we utilize in the proposed algorithms is a (computationally) privacy preserving intersection of deterministic finite automata (DFA) (Guanciale, Gurov, and Laud 2014).

In PP-MAP literature, two notions of privacy formulated in (Brafman 2015) prevail. The first notion is *weak privacy*, which requires the agents not to communicate private actions and private parts of state and public actions directly, that is, without encryption. This notion is assumed by a significant portion of PP-MAP planners (see (Komenda, Stolba, and Kovacs 2016) for a comprehensive overview), but does not provide any real privacy guarantees, as information can be deduced from the communicated (although encrypted) information. The second notion is *strong privacy*, which coincides with the cryptographic and MPC definitions of privacy and privacy preserving computation. That is, in strong privacy-preserving MAP, no information can leak from the computation, apart form what can already be deduced from the public part of the input ($\mathcal{M}^{\triangleright}$) and of the output ($\pi^{\triangleright}$). No MAP planner strong privacy preserving on all MAP problems has been proposed in the literature up to date.

## A Strong Privacy Preserving Planner

In this section, we present a class of PP-MAP algorithms, based on the PSM structure[1] and the generate and test paradigm. The generic structure of a PSM-based planner is listed in Algorithm 1.

If $S_i$ is finite, PSI can be used instead of a DFA intersection. By instantiating each step of this scheme we create several types of PSM planners:

**Algorithm 1:** Generic PSM-based Planner

**Algorithm** `GenericPSM(`$\mathcal{M}$`)`

1. Each agent $i \in \{1, ..., n\}$ generates a set $S_i$ of local solutions of $\Pi_i$, stored in a PSM $\Gamma(S_i)$.
2. Each agent $i$ computes a public projection $\Gamma(S_i)^{\triangleright}$.
3. All agents compute together the intersection $\bigcap_{i=1}^{n} \Gamma(S_i)^{\triangleright}$ using a secure DFA intersection.
4. If $\bigcap_{i=1}^{n} \Gamma(S_i)^{\triangleright} \neq \emptyset$, the intersection represents a nonempty set of global solutions to $\mathcal{M}$, continue with Step 5. Otherwise, either terminate and report no solution, or continue with Step 1.
5. Jointly and securely select one random solution from $\bigcap_{i=1}^{n} \Gamma(S_i)^{\triangleright}$.

**One-shot-PSM planner** generates a proper random subset of all solutions in Step 1. and terminates in Step 4. if a solution is not found.

**Iterative-PSM planner** repeats all steps until the intersection $\bigcap_{i=1}^{n} \Gamma(S_i)^{\triangleright}$ is nonempty, or all agents have constructed a full PSM $\Gamma(\Pi_i)$, in which case if the intersection is empty, there is no solution. In each iteration of Step 1., new plans are added systematically (e.g., ordered by length).

**Full-PSM planner** each agents constructs a full PSM $\Gamma(\Pi_i)$ in Step 1. If the problem has a solution, all solutions are found in the first iteration of Step 4.

The Iterative-PSM and Full-PSM planners were already published in (Tozicka et al. 2016), albeit without the use of secure DFA intersection, whereas One-shot-PSM is a novel variant of the planner. Notice that both One-shot-PSM and Full-PSM planners are computationally strong privacy preserving as the secure DFA intersection by (Guanciale, Gurov, and Laud 2014) is computationally strong privacy preserving and no other communication is performed. Also, in the case of One-shot-PSM, an information-theoretic PSI (Li and Wu 2007) can be used as the used sets of plans can be finite and thus One-shot-PSM can be strong privacy preserving in the information-theoretic sense (without any assumptions). Another promising feature of the One-shot-PSM planner is that there is a trade-off between completeness and efficiency, which can be exploited. The more plans are generated, the more time it takes, but also the higher is the chance of success in the one shot secure PSM intersection.

Before formulating these observations formally, we provide an algorithm for secure selection of a random solution from the intersection of PSMs, as is required in the Step 5.

### Random Solution Selection

In (Guanciale, Gurov, and Laud 2014) the authors propose algorithm for secure intersection of regular languages,

[1] The PSM structure and a planner based on it was originally published in (Tozicka et al. 2016). The use of secure DFA intersection is novel as well is the One-shot-PSM planner.

which is based on DFA minimization, secure intersection and secure trimming of unreachable states. As PSM is an instance of DFA, this techniques can be used also to securely compute intersection of the agents' PSMs and securely remove unreachable states, that is, obtain a minimal DFA representing the resulting PSM. The next step we need to perform is to select a random solution, again, without leaking private information (Step 5. of Algorithm 1).

To select a random solution (a public plan) from the intersection of PSMs $\bigcap_{i=1}^{n} \Gamma(S_i)^{\triangleright}$, we will iteratively select a random transition (action) from $\delta$ of $\bigcap_{i=1}^{n} \Gamma(S_i)^{\triangleright}$ leading from its initial state $s_I$ through intermediate states $s \in N$ and eventually terminates in one of the terminals in $F$ (the goal states). In states where we can both continue with a transition from $\delta$ or terminate, we will chose randomly whether to continue with one of the randomly selected transitions or whether we will terminate. The resulting random trace through the PSM will be in form $(\mathsf{lbl}(a_1) \in \Sigma, \ldots, \mathsf{lbl}(a_k) \in \Sigma)$. The extracted plan is then straightforwardly $\pi = (a_1, \ldots, a_k)$.

All constructs in this procedure can be implemented with existing MPC techniques (e.g., Sharemind (Bogdanov, Laur, and Willemson 2008)). The iterative concatenation of the trace (which is the public plan) as well as the number of iterations does not have to be hidden as it is part of the public output. The used random variables have all uniform distribution which does not reveal any additional information as well. The procedure can work only with minimal DFA (in a non-minimal DFA it could randomly end up in a state which is not a terminal and there is no outgoing transition from it), which holds for the intersection $\bigcap_{i=1}^{n} \Gamma(S_i)^{\triangleright}$ of the PSMs.

## Example

Let us show how the presented PSM-based algorithms work on a simple case of a coalition surveillance mission problem with one UAV and two secret locations (see Figure 1), where UAVs survey an area and need to be refueled by coalition partners (a coalition base), the surveyed areas and the state of supplies of the base are private (secret). We omit the movement actions for simplicity (movement between the surveyed locations would be private, movement to the coalition base would be public).

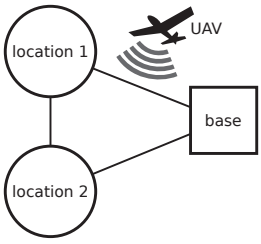


Figure 1: UAV surveillance scenario example.

The problem in the running example consists of two agents $\mathcal{A} = \{\mathsf{UAV}, \mathsf{base}\}$ and can be described using the following sets of propositions:

| Description | Proposition | $s_I$ | $s_\star$ |
|---|---|---|---|
| UAV has fuel | $\mathsf{f} \in P^{\mathsf{pub}}$ | false | - |
| mission is complete | $\mathsf{c} \in P^{\mathsf{pub}}$ | false | true |
| location 1 is complete | $\mathsf{l1} \in P^{\mathsf{priv}}_{\mathsf{UAV}}$ | false | - |
| location 2 is complete | $\mathsf{l2} \in P^{\mathsf{priv}}_{\mathsf{UAV}}$ | false | - |
| base has enough supplies | $\mathsf{s} \in P^{\mathsf{priv}}_{\mathsf{base}}$ | true | - |

The problem consists of the following actions:

| Actions (UAV) | $\mathsf{lbl}(a)$ | $\mathsf{pre}(a)$ | $\mathsf{add}(a)$ | $\mathsf{del}(a)$ |
|---|---|---|---|---|
| survey location 1 | SL1 | $\{\mathsf{f}, \neg\mathsf{l1}\}$ | $\{\mathsf{l1}\}$ | $\{\mathsf{f}\}$ |
| survey location 2 | SL2 | $\{\mathsf{f}, \neg\mathsf{l2}\}$ | $\{\mathsf{l2}\}$ | $\{\mathsf{f}\}$ |
| complete mission | C | $\{\mathsf{l1}, \mathsf{l2}\}$ | $\{\mathsf{c}\}$ | $\emptyset$ |

| Actions (base) | $\mathsf{lbl}(a)$ | $\mathsf{pre}(a)$ | $\mathsf{add}(a)$ | $\mathsf{del}(a)$ |
|---|---|---|---|---|
| refuel | R | $\{\neg\mathsf{f}, \mathsf{s}\}$ | $\{\mathsf{f}\}$ | $\{\mathsf{s}\}$ |
| refuel, resupply | RR | $\{\neg\mathsf{f}, \neg\mathsf{s}\}$ | $\{\mathsf{f}, \mathsf{s}\}$ | $\emptyset$ |

The public projection of the problem is restricted to the public propositions $P^{\mathsf{pub}}$ and the public projections of the actions. Note that as SL1 and SL2 have the same projection, they cannot be distinguished and thus are represented by a single projected action $\mathsf{SL}^{\triangleright}$. The same holds for the actions R and RR which are both represented by a projected action $\mathsf{R}^{\triangleright}$. The details of the projected actions are the following:

| Actions ($\mathsf{UAV}^{\triangleright}$) | | $\mathsf{pre}(a^{\triangleright})$ | $\mathsf{add}(a^{\triangleright})$ | $\mathsf{del}(a^{\triangleright})$ |
|---|---|---|---|---|
| survey location | $\mathsf{SL}^{\triangleright}$ | $\{\mathsf{f}\}$ | $\emptyset$ | $\{\mathsf{f}\}$ |
| compl. mission | $\mathsf{C}^{\triangleright}$ | $\emptyset$ | $\{\mathsf{c}\}$ | $\emptyset$ |

| Actions ($\mathsf{base}^{\triangleright}$) | | $\mathsf{pre}(a^{\triangleright})$ | $\mathsf{add}(a^{\triangleright})$ | $\mathsf{del}(a^{\triangleright})$ |
|---|---|---|---|---|
| refuel | $\mathsf{R}^{\triangleright}$ | $\{\neg\mathsf{f}\}$ | $\{\mathsf{f}\}$ | $\emptyset$ |

Full PSMs and their projections for both agents are shown in Figure 2. Their intersection equals to the full PSM of the UAV agent. That is how the *Full-PSM planner* works. On the other hand, when using *One-shot-PSM planner*, the base agent can decide to add only one local solution to its PSM, namely the local plan $\{\mathsf{R}, \mathsf{C}\}$. In that case, the intersection of agents' PSMs is empty and thus the planner ends without finding a solution. Of course, even in *One-shot-PSM planner*, both agents may add multiple solutions and find a global solution.

In the case of *Iterative-PSM planner*, the base agent adds another plan $\{\mathsf{C}\}$ to its PSM, which still yields an empty intersection. At third iteration, the base agent adds also $\{\mathsf{R}, \mathsf{SL}, \mathsf{C}\}$ to its PSM and thus represents all necessary plans. Intersection of such PSM with the UAV PSM is non-empty and contains solution of the problem. In this case, the base agent knows that the UAV agent does not accept plan $\{\mathsf{R}, \mathsf{C}\}$, which leaks private information. Note that the base agent cannot deduce this information in the cases of *Full-PSM planner* and *One-shot-PSM planner* because the intersection of PSMs is not known to the agents and the final solution is selected randomly from all solutions encoded by the PSM intersection.

## The Limits of Strong Privacy Preserving MAP

In this section, we present theoretical limits of the privacy preserving planner described above and their generalization to other PP-MAP paradigms. For the privacy analysis, we

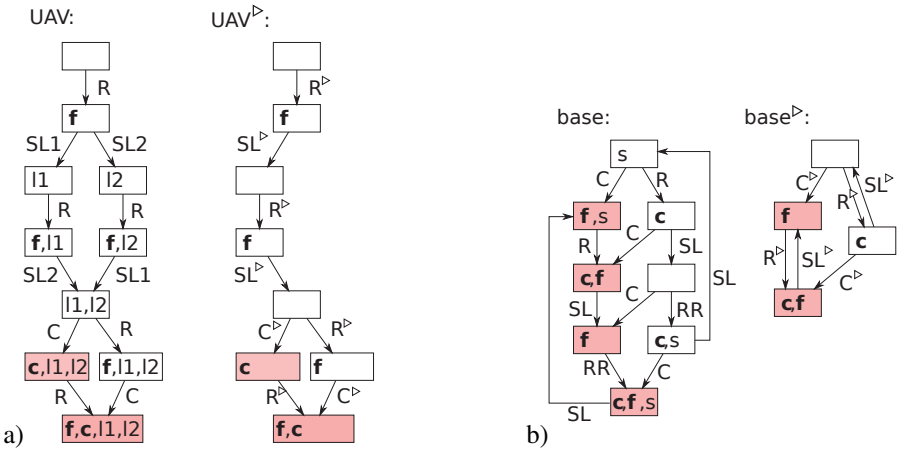


Figure 2: a) Full PSM of the UAV agent and its public projection. b) Full PSM of the base agent and its public projection.

assume that the agents know the algorithm used by all other agents (also including probability distributions of any random variables, e.g., the uniform distributions in the random solution selection) and also that the MAP problems cannot be solved by a single agent alone (in which case a strong privacy-preserving algorithm is trivial). We focus on the following three properties

**Definition 2.** A MAP planner $P$ is

**(i) Complete** if for each MAP problem $\mathcal{M}$ that has a solution (a global plan), $P$ terminates and returns a solution to $\mathcal{M}$.

**(ii) Strong privacy preserving** if $P$ does not reveal any other information than what can be deduced from the public part of the input and the solution, that is, $\mathcal{M}^{\triangleright}$ and $\pi^{\triangleright}$.

**(iii) Efficient** if there exist a MAP problem $\mathcal{M} = \{\Pi_i\}_{i=1}^n$ for which $P$ always returns a solution without enumerating all public solutions of each $\Pi_i$.

The completeness definition does not require any further explanation and the privacy definition has already been discussed. As already mentioned, based on the presence of computational assumptions, we distinguish two flavors of strong privacy preserving algorithms, computational and information-theoretic. The efficiency definition is somewhat unusual. The aim is to differentiate between algorithms which do have to explore the complete private search spaces of the agents (which, in the worst case, can be as big as the global problem) and those which do not. Although the theoretical complexity class is the same for both, as the worst case is always the complete exploration, in most practical problems, the difference is significant.

## The Limits of the PSM-based Planners

i In this section, we assign the properties from Definition 2 to the particular PSM-based planner variants.

**Theorem 3.** *The Full-PSM planner is* complete *and computationally* strong privacy preserving.

*Proof.* Let $\mathcal{M} = \{\Pi_i\}_{i=1}^n$ be a MAP problem. The Full-PSM generates a full PSM $\Gamma(\Pi_i)$ for each $\Pi_i$ and a public projection $\Gamma(\Pi_i)^{\triangleright}$, each representing all local solutions of each respective $\Pi_i$ and their public projection. Then a PSM intersection $\bigcap_{i=1}^n \Gamma(\Pi_i)^{\triangleright}$ is computed. If a global solution $\{\pi_i\}_{i=1}^n$ to $\mathcal{M}$ exists, each $\pi_i$ is a local solution to $\Pi_i$ and thus is represented by $\Gamma(\Pi_i)$. Because $\pi_i^{\triangleright} = \pi_j^{\triangleright}$ for each $i, j$, $\pi_i^{\triangleright}$ is represented by the intersection $\bigcap_{i=1}^n \Gamma(\Pi_i)^{\triangleright}$ and is a public projection of a global solution, privately extensible by all agents. Thus Full-PSM is complete.

If the intersection $\bigcap_{i=1}^n \Gamma(\Pi_i)^{\triangleright}$ is computed using a privacy preserving DFA intersection by (Guanciale, Gurov, and Laud 2014), Full-PSM is computationally strong privacy preserving as no other multiparty computation or communication is performed and no other information is exchanged. □

**Corollary 4.** *The Full-PSM planner is not* efficient.

*Proof.* Trivial, as the Full-PSM from definition always generates local solutions of all agents before computing the intersection. □

**Theorem 5.** *The Iterative-PSM planner is* complete *and* efficient.

*Proof.* Let $\mathcal{M} = \{\Pi_i\}_{i=1}^n$ be a MAP problem and let $\{\pi_i\}_{i=1}^n$ be a global solution to $\mathcal{M}$. Even though the number of all possible solutions to $\mathcal{M}$ may be infinite, each such solution $\{\pi_i\}_{i=1}^n$ is finite and has a length $l$. As Iterative-PSM is adding the plans in a systematic way (that is, a plan of length $k+1$ is added only after all plans of length $k$ were added), all plans of length $l$ are added to each $S_i$ after a finite number of steps. Thus also all the plans in $\{\pi_i\}_{i=1}^n$ are added after a finite many steps and the solution becomes part of the intersection $\bigcap_{i=1}^n \Gamma(S_i)^{\triangleright}$ and thus Iterative-PSM is complete.

Let $l$ be the length of the shortest global solution. Based on the systematic generation described above, the solution is always found before all solutions of length $l' > l$ and thus Iterative-PSM is efficient according to Definition 2(iii). □

**Theorem 6.** *The Iterative-PSM planner is not* strong privacy preserving.

*Proof.* By iterating the PSI or secure DFA intersection, information is leaked. In particular, information that a plan $\pi_i$ shorter than the solution proposed by agent $i$ is not extensible by some agent $j \neq i$. This reveals the existence of private preconditions of some public actions of agent $j$ used in $\pi_i$. Note that as we assume the knowledge of the algorithm by all agents, even a less obvious systematic generation of plans leaks information as the particular algorithm can be simulated by other agents and the plans which should have already been generated can be determined. In the case of randomized algorithms, we assume the knowledge of the probability distributions of random variables used in the algorithm as part of the algorithms and thus again, the information about unaccepted public plans leaks. □

**Theorem 7.** *The One-shot-PSM planner is* strong privacy preserving *and* efficient*.*

*Proof.* By computing the secure DFA intersection only once, no additional information can leak and thus One-shot-PSM is computationally strong privacy preserving. If each PSM $\Gamma(S_i)$ is replaced by a finite subset of represented plans $S_i$, the PSI can be used instead of the DFA intersection. By using an information theoretic secure PSI (Li and Wu 2007) on finite sets of plans, One-shot-PSM becomes information-theoretic strong privacy preserving.

One-shot-PSM is trivially *efficient* according to Definition 2(iii) as it can use arbitrarily small subsets of all possible local solutions. □

**Theorem 8.** *The One-shot-PSM planner is* not complete.

*Proof.* As some public solution is not generated by at least one of the agents, it may be the case that the not-generated solution is the one and only solution of the problem and thus such problem would not be solved. □

## Impossibility Theorem

Next, we state the main contribution of this paper for the class of PSM-based planners and later generalize it to a wider class of planning algorithms.

**Theorem 9.** *A PSM-based MAP planner $P$ cannot have all three properties (Definition 2)* complete*,* strong privacy preserving *and* efficient *together.*

*Proof.* According to Theorem 3, the Full-PSM is complete and strong privacy preserving, but according to Corollary 4 not efficient as it generates all local solutions. For the sake of contradiction, let us have a complete and strong privacy preserving planner $P$ which is efficient. From Definition 2(iii) follows that there exist a MAP problem $\mathcal{M} = \{\Pi_i\}_{i=1}^n$ for which some of the agents using $P$ do not have to generate all public plans in order to find a global plan $\{\pi_i\}_{i=1}^n$, let $j$ be such agent. Let $\bar{\pi}_j^{\triangleright} \neq \pi_j^{\triangleright}$ be the public plan which is not generated by agent $j$.

Because we assume that the problem $\mathcal{M}$ cannot be solved by a single agent only, a MAP problem $\bar{\mathcal{M}}$ can be constructed from $\mathcal{M}$ so that the only public plan extensible by all agents is $\bar{\pi}_j^{\triangleright}$. It is enough, if one of the agents rejects all public plans not equal to the public plan $\bar{\pi}_j^{\triangleright}$ an therefore the newly constructed MAP problem $\bar{\mathcal{M}}$ can differ from $\mathcal{M}$ only in the problem of one agent, let that be agent $i$. The construction is as follows. Let $\bar{\pi}_i$ be a local plan of agent $i$ such that $\bar{\pi}_i^{\triangleright} = \bar{\pi}_j^{\triangleright}$, that is, $\bar{\pi}_i$ can be part of the global plan as it is the extension of $\bar{\pi}_j^{\triangleright}$ by agent $i$.

We construct $\bar{\Pi}_i$ from $\Pi_i = \left\langle P_i = P^{\mathsf{pub}} \cup P_i^{\mathsf{priv}}, A_i, s_I \cap P_i, s_\star \cap P_i \right\rangle$ by adding a new proposition $p_k$ for each public action $a_k \in \bar{\pi}_i^{\triangleright}$ s.t. $a_k \in A_i$ and by adding a new proposition $p_{\mathsf{neg}}$. We add $p_0$ to $s_I$ and modify each such $a_k$ so that $\mathsf{pre}(\bar{a}_k) = (\mathsf{pre}(a_k) \cap P^{\mathsf{pub}}) \cup \{p_k\}$, $\mathsf{add}(\bar{a}_k) = (\mathsf{pre}(a_k) \cap P^{\mathsf{pub}}) \cup \{p_{k+1}\}$ or $\mathsf{add}(\bar{a}_k) = \mathsf{pre}(a_k) \cap P^{\mathsf{pub}}$ if $a_k$ is the last action and $\mathsf{del}(\bar{a}_k) = \mathsf{del}(a_k) \cap P^{\mathsf{pub}}$. We modify each $a'_k \in A_i$ s.t. $a_k \notin \bar{\pi}_i^{\triangleright}$ so that $\mathsf{pre}(\bar{a}_k) = (\mathsf{pre}(a_k) \cap P^{\mathsf{pub}}) \cup \{p_{\mathsf{neg}}\}$. The result is that only actions in $\bar{\pi}_i$ are applicable and only in the exact same order, also keeping the public constraints in place, thus $\{\bar{\pi}_i\}_{i=1}^n$ is the only global solution to $\bar{\mathcal{M}}$.

Since $P$ is strong privacy preserving and $\mathcal{M}^{\triangleright} = \bar{\mathcal{M}}^{\triangleright}$ as the public part was not modified, the agent $j$ cannot distinguish between $\mathcal{M}$ and $\bar{\mathcal{M}}$ and thus generates exactly the same PSMs $\Gamma(\Pi_j) = \Gamma(\bar{\Pi}_j)$ for both problems. But then, as the planner $P$ is complete and $\bar{\mathcal{M}}$ has the only solution $\{\bar{\pi}_i\}_{i=1}^n$, the agent $j$ has to generate $\bar{\pi}_j$ also for $\mathcal{M}$. Thus we obtain a contradiction with the assumption that the planner $P$ is efficient (because it has to also generate $\bar{\pi}_j$), in other words that a strong privacy preserving and complete planner can generate less local plans than Full-PSM which generates all of them and thus violates the efficiency property according to Definition 2(iii). □

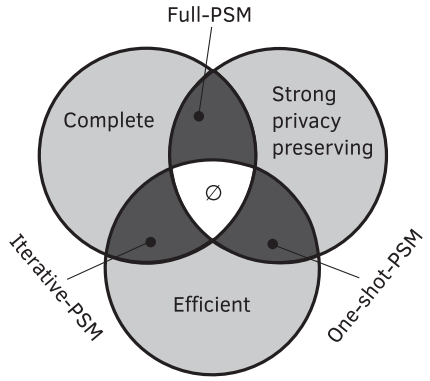


Figure 3: Properties of PSM-based planners.

To illustrate the above proof, we will modify the UAV example. Let UAV be the agent $j$ and let $\pi_{\mathsf{UAV}}^{\triangleright} = \{\mathsf{SL}, \mathsf{R}, \mathsf{SL}, \mathsf{R}, \mathsf{C}\}$ be the public plan which is not generated by the UAV agent. Let the corresponding local plan of the base agent be $\pi_{\mathsf{base}} = \{\mathsf{SL}, \mathsf{R}, \mathsf{SL}, \mathsf{RR}, \mathsf{C}\}$. Then the problem of the base agent can be modified so that $P_{\mathsf{base}}^{\mathsf{priv}} = \{p_1, p_2, p_{\mathsf{neg}}\}$ and each action in the plan is modified so that $\mathsf{pre}(\mathsf{R}) = \{\neg\mathbf{f}, p_1\}$, $\mathsf{add}(\mathsf{R}) = \{\mathbf{f}, p_2\}$, $\mathsf{pre}(\mathsf{RR}) = \{\neg\mathbf{f}, p_2\}$ and if there was any other action of the agent base, its private

preconditions would be set to $\{p_{\mathsf{neg}}\}$. Also, the private part of $s_I$ is set to $\{p_1\}$ and thus only the action R is applicable.

The properties of PSM-based planners according to Definition 2 and the above theorems are summarized in Figure 3. Notice that the intersection of all properties is empty, as shown by the Theorem 9. Also, the Full-PSM and One-shot-PSM are the only generally strong privacy preserving planners published up to date, thanks to the novel use of the secure DFA intersection.

Notice that the Theorem 9 holds not only for the PSM Planner (Tozicka et al. 2016) and the family of PSM-based planners described in Algorithm 1, but also for algorithms based on similar generate and test paradigm, such as Planning First (Nissim, Brafman, and Domshlak 2010), which uses distributed constraint satisfaction (DCSP) in the place of PSM intersection. The potential use of a secure version of DCSP would result in similar limits as are the limits of the PSM planner.

## The Limits of State-Space Search

In this section, we focus on the strong privacy preserving property in general terms. We abstract from the particular secure computations such as PSI, secure DFA intersection or even a single ORAM computation and refer to them collectively as secure primitives. The important property of a secure primitive is that (possibly under computational security assumptions), no information leaks from a single secure primitive and thus, on its own, it is *strong privacy preserving*. In general, combining multiple secure primitives may leak information, as shown e.g., in Theorem 6. Formally:

**Definition 10. (Secure primitive)** A secure primitive (or a cryptographic primitive) is a (possibly multiparty) computation block which by itself is strong privacy preserving.

We first state two general results applicable to any MAP planning algorithm and then use them to generalize Theorem 9 to wider classes of MAP planners. The definitions are based on the notion of publicly equivalent MAP problems:

**Definition 11.** Two MAP problems $\mathcal{M}$, $\mathcal{M}'$ are publicly equivalent if $\mathcal{M}^{\triangleright} = \mathcal{M}'^{\triangleright}$ and their respective sets of public solutions $S^{\triangleright}$ and $S'^{\triangleright}$ are equal, formally $\mathcal{M} \equiv \mathcal{M}'$.

**Theorem 12.** *Let $P$ be a MAP planner and $\mathcal{M}$, $\mathcal{M}'$ two publicly equivalent MAP problems $\mathcal{M} \equiv \mathcal{M}'$, such that $\mathcal{M}$ and $\mathcal{M}'$ differ in the private part of agent $i$. Then if $P$ is strong privacy preserving, it performs the same number of secure primitives on both $\mathcal{M}$ and $\mathcal{M}'$.*

*Proof.* Trivially, if $P$ performs different number of secure primitives on $\mathcal{M}$ and $\mathcal{M}'$, the agents other than $i$ can distinguish between $\mathcal{M}$ and $\mathcal{M}'$, which is an information that cannot be learned from the output, as the public projections of both solutions are equal. □

**Corollary 13.** *The number of secure primitives performed by a strong privacy preserving MAP planner $P$ cannot depend on any private part of the MAP problem $\mathcal{M}$.*

*Proof.* Direct consequence of Theorem 12. □

Note that the whole planning algorithm can be considered a secure primitive, if it is strong privacy preserving, e.g., the Full-PSM planner.

The Theorem 12 and Corollary 13 are very general, but also provide necessary conditions for any strong privacy preserving MAP planner. This conditions can be used to generalize the Theorem 9 to a wider class of MAP algorithms.

**Definition 14.** A state-space search (SS) MAP planner is a MAP planner in which each agent searches its own state-space. The agents coordinate themselves by exchanging public projections of reachable states.

An example of a SS MAP planner aiming for secure computation is Secure-MAFS (Brafman 2015) which is strong privacy preserving for a restricted class of problems.

**Corollary 15.** *Theorem 9 holds for any SS MAP planner, assuming a bound $b$ on the number of states in the global state space of $\mathcal{M}$.*

*Proof.* Let us assume that the states in a strong privacy preserving SS MAP planner $P$ are communicated in a secure way, that is, no information is leaked by communicating a single state and thus the communication of a single state can be considered a secure primitive. According to Corollary 13, the number $p$ of secure primitives must depend only on the public part of $\mathcal{M}$. In order to be *complete*, $p$ must be large enough even for the worst case execution, which is if the state space is of size $b$ and all states are expanded and sent. But this corresponds to enumerating all local solutions of $\mathcal{M}$ and thus breaks the *efficiency* property according to Definition 2(iii). □

Based on (Stolba, Tozicka, and Komenda 2016a), the forward-chaining plan-space search as used in FMAP (Torreño, Onaindia, and Sapena 2014) essentially corresponds to the state-space search paradigm and thus the same results apply.

As already mentioned, the question whether a generic MPC technique such as ORAM or BlindTM can be used for efficient strong privacy preserving MAP planning is left open for the future work. This concludes the theoretical analysis of the limits of strong privacy preserving multi-agent planning, both in general and in particular case of the PSM-based planners.

# Strong Privacy Preserving Equivalence Classes

In (Brafman 2015), the author proves that Secure MAFS is strong privacy preserving on a restricted class of logistics problems, where the problems have the same set of packages, the same set of public locations, and identical initial public locations for packages and also that every private location is reachable from every other private location. This effectively means that part of the private problem is irrelevant (that is, it can always be solved) and the rest of the private problem is the same for all instances of the restricted problem. In this section, we formalize the notion of privacy used in the proof in (Brafman 2015) and generalize the idea of privacy on a class of problems.

**Definition 16.** A MAP planner $P$ is *strong privacy preserving on a class* $C$ of MAP problems iff from the execution of $P$ on $\mathcal{M} \in C$ and on $\mathcal{M}' \in C$ no agent can distinguish $\mathcal{M}$ and $\mathcal{M}'$.

This definition of strong privacy differs from that in Definition 2, but is reasonable. If solving e.g., a logistics problem, even if the fact that a package is loaded is private, the agents can expect its existence based on how logistics works. Now we formalize the equivalence class of problems based on Definition 11.

**Definition 17.** A class $C$ of MAP problems is a *public equivalence class* iff for each two $\mathcal{M} \in C$ and $\mathcal{M}' \in C$ holds $\mathcal{M} \equiv \mathcal{M}'$.

This means that $\mathcal{M}^{\triangleright} = \mathcal{M}'^{\triangleright}$ and their respective sets of public solutions $S^{\triangleright}$ and $S'^{\triangleright}$ are equal. For each of the problems $\mathcal{M} \in C$ by itself, the private part of $\mathcal{M}$ poses a constraint and thus reduces the number of public solutions which are also global solutions. But as all problems in $C$ have equal public projection and also the set of public solutions, each of the problems in $C$ differ from the other problems only by such part of the private problem, which does not add more constraints and prevent more solutions, that is, the different private parts of the problems can always be solved.

An example of such class $C$ are the logistics problems used in (Brafman 2015) and rephrased at the beginning of this section. The particular problems differ by the actual number of private locations, but as the private locations are always connected (not necessarily directly) to a public location, this part of the problem does not constraint the public solutions which are also global solutions of the whole problem. Nevertheless, this does not mean that the private part of $\mathcal{M}$ is unnecessary—the agents still need to cooperate in order to solve $\mathcal{M}$ and some of the public solutions are not extensible because of the private parts of the agent problems.

Based on the Definition 17, we can formulate a general result.

**Theorem 18.** *A MAP planner $P$ which is complete and efficient by Definition 2 and strong privacy preserving on a public equivalence class $C$ of MAP problems by Definition 16 exists.*

*Proof.* Let us start with Iterative-PSM, which is complete and efficient by Theorem 5. In each iteration, a plan $\pi_i$ proposed by agent $i$ is either accepted by all other agents, in which case the algorithm ends and no information leaks, because $\pi_i$ is part of the solution, or $\pi_i$ is not accepted and thus some information leaks. But in the case of public equivalence class $C$ of problems, $\pi_i$ is either accepted or not accepted in all $\mathcal{M} \in C$ and thus this information cannot be used to distinguish any two $\mathcal{M}, \mathcal{M}' \in C$. Therefore by Definition 16, Iterative-PSM is strong privacy preserving on the class $C$. □

This theorem generalizes the results of (Brafman 2015) to all public equivalence classes of MAP problems and also to MAP planners in general. Moreover, we can formulate the following corollary.

**Corollary 19.** *The public equivalence relation $\equiv$ partitions all MAP problems into classes of equivalence. There exists a MAP planner $P$, which is complete and efficient by Definition 2 and for each MAP problem $\mathcal{M}$, $P$ is strong privacy preserving on a public equivalence class $C$ of MAP problems, induced by $\mathcal{M}$.*

This means that each MAP problem $\mathcal{M}$ induces the class $C$ of publicly equivalent problems, which can be solved by such planner $P$ (e.g., the Iterative-PSM planner), revealing no other information, than that the problem falls in the particular class $C$. It seems that for some practical applications, this might be enough to consider the planner strong privacy-preserving, as the participating agents already know the class of the planning problems they are solving in advance (e.g., the logistics problems with particular constraints).

Considering the UAV example, a problem $\mathcal{M}'$ which does not consider the supplies of the base agent (there is no private proposition) $s$ falls in the same equivalence class as the original problem as the RR can always be used and provide supplies. Also, all problems which have more complex private parts (e.g., another private actions for preparing the fuel, etc.) which do not restrict any solutions of the original problem fall in the same equivalence class $C$.

## Conclusions and Future Work

In this paper, we have extended an existing planner PSM, the winner of the coverage and quality distributed tracks of the CoDMAP competition, with a strong privacy preserving protocol which together allows strong privacy preserving multi-agent planning. We have shown that it is not possible for such planner to be strong privacy preserving, complete and efficient at the same time and we have proposed three variants of the planner which satisfy each two of the mentioned properties. For practical use, the One-shot PSM planner is the most suitable as it is both strong privacy preserving and efficient, although incomplete. Moreover, there is a trade-off between completeness and efficiency as the more plans are generated ahead of the secure primitive, the higher is the chance of success.

We have also generalized the result that it is not possible for a MAP planner to be strong privacy preserving, complete and efficient at the same time, to a much wider class of state-space search based MAP planners. Additionally, we have provided a new notion of privacy restricted to a class of problems, where it is easier to obtain strong privacy and where it is possible to satisfy all three properties of completeness, strong privacy and efficiency at once.

We propose two directions of future work. The first is to assess the real performance of the One-shot PSM planner and the price it pays for strong privacy experimentally. The second direction of future work is to investigate the use of general-purpose secure computations such as ORAM and Blind Turing Machine, which might possibly be used for strong privacy preserving multi-agent planning, but with a yet unknown overhead.

**Acknowledgments** We highly appreciate the valuable feedback from the anonymous reviewers. This research was

supported by the Czech Science Foundation (grant no. 15-20433Y) and by the Grant Agency of the CTU in Prague (grant no. SGS14/202/OHK3/3T/13).

## References

Bogdanov, D.; Laur, S.; and Willemson, J. 2008. Sharemind: A framework for fast privacy-preserving computations. In *European Symposium on Research in Computer Security*, 192–206. Springer.

Boyle, E., and Naor, M. 2016. Is there an oblivious RAM lower bound? In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, 357–368. ACM.

Brafman, R. I., and Domshlak, C. 2008. From one to many: Planning for loosely coupled multi-agent systems. In *Proceedings of the 18th International Conference on Automated Planning and Scheduling, ICAPS*, 28–35.

Brafman, R. I. 2015. A privacy preserving algorithm for multi-agent planning and search. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI*, 1530–1536.

Brickell, J., and Shmatikov, V. 2005. Privacy-preserving graph algorithms in the semi-honest model. In *International Conference on the Theory and Application of Cryptology and Information Security*, 236–252. Springer.

Goldreich, O. 1987. Towards a theory of software protection and simulation by oblivious rams. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, 182–194. New York, NY, USA: ACM.

Guanciale, R.; Gurov, D.; and Laud, P. 2014. Private intersection of regular languages. In *Privacy, Security and Trust (PST), 2014 Twelfth Annual International Conference on*, 112–120. IEEE.

Jarecki, S., and Liu, X. 2010. Fast secure computation of set intersection. In *Security and Cryptography for Networks, 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010. Proceedings*, 418–435.

Komenda, A.; Stolba, M.; and Kovacs, D. L. 2016. The international competition of distributed and multiagent planners (CoDMAP). *AI Magazine* 37(3):109–115.

Li, R., and Wu, C. 2007. An unconditionally secure protocol for multi-party set intersection. In *Applied Cryptography and Network Security*, 226–236. Springer.

Maliah, S.; Brafman, R. I.; and Shani, G. 2016. Increased privacy with reduced communication and computation in multi-agent planning. In *Proceedings of the 4th Workshop on Distributed and Multi-Agent Planning, DMAP–ICAPS'16*, 106–112.

Maliah, S.; Shani, G.; and Brafman, R. I. 2016. Online macro generation for privacy preserving planning. In *Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016.*, 216–220.

Maliah, S.; Shani, G.; and Stern, R. 2016. Stronger privacy preserving projections for multi-agent planning. In *Proceedings of the Twenty-Sixth International Conference on Automated Planning and Scheduling, ICAPS 2016, London, UK, June 12-17, 2016.*, 221–229.

Nissim, R., and Brafman, R. I. 2014. Distributed heuristic forward search for multi-agent planning. *J. Artif. Intell. Res. (JAIR)* 51:293–332.

Nissim, R.; Brafman, R. I.; and Domshlak, C. 2010. A general, fully distributed multi-agent planning algorithm. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, 1323–1330. International Foundation for Autonomous Agents and Multiagent Systems.

Pinkas, B.; Schneider, T.; Segev, G.; and Zohner, M. 2015. Phasing: Private set intersection using permutation-based hashing. In *24th USENIX Security Symposium (USENIX Security 15)*, 515–530. Washington, D.C.: USENIX Association.

Rass, S.; Schartner, P.; and Brodbeck, M. 2015. Private function evaluation by local two-party computation. *EURASIP J. Information Security* 2015:7.

Stolba, M.; Tozicka, J.; and Komenda, A. 2016a. Secure multi-agent planning. In *Proceedings of the 1st International Workshop on AI for Privacy and Security, PrAISe@ECAI 2016, The Hague, Netherlands, August 29-30, 2016*, 11:1–11:8.

Štolba, M.; Tožička, J.; and Komenda, A. 2016b. Secure multi-agent planning algorithms. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI'16)*, 1714–1715.

Torreño, A.; Onaindia, E.; and Sapena, O. 2014. FMAP: distributed cooperative multi-agent planning. *Appl. Intell.* 41(2):606–626.

Tozicka, J.; Jakubuv, J.; Komenda, A.; and Pechoucek, M. 2016. Privacy-concerned multiagent planning. *Knowl. Inf. Syst.* 48(3):581–618.

Yao, A. C. 1982. Protocols for secure computations. In *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science, SFCS*, 160–164.

Yao, A. C.-C. 1986. How to generate and exchange secrets. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, SFCS '86, 162–167. Washington, DC, USA: IEEE Computer Society.