Towards a General Theory of Non-Cooperative Computation

(Extended Abstract)

Robert McGrew, Ryan Porter, and Yoav Shoham
Stanford University
{bmcgrew,rwporter,shoham}@cs.stanford.edu

Abstract

We generalize the framework of non-cooperative computation (NCC), recently introduced by Shoham and Tennenholtz, to apply to cryptographic situations. We consider functions whose inputs are held by separate, self-interested agents. We consider four components of each agent's utility function: (a) the wish to know the correct value of the function, (b) the wish to prevent others from knowing it, (c) the wish to prevent others from knowing one's own private input, and (d) the wish to know other agents' private inputs. We provide an exhaustive game theoretic analysis of all 24 possible lexicographic orderings among these four considerations, for the case of Boolean functions (mercifully, these 24 cases collapse to four). In each case we identify the class of functions for which there exists an incentive-compatible mechanism for computing the function. In this article we only consider the situation in which the inputs of different agents are probabilistically independent.

1 Introduction

In this paper we analyze when it is possible for a group of agents to compute a function of their privately known inputs when their own self-interests stand in the way. One motivation for studying this class of problems is cryptography. Consider, for example, the problem of secure function evaluation (SFE). In SFE, n agents each wish to compute a function of n inputs (where each agent i possesses input i), without revealing their private inputs. An increasingly clever series of solutions to SFE have been proposed (e.g., [1, 2]). But if these protocols are the answer, what exactly is the question? Like many other cryptographic problems, SFE has not been given a mathematical definition that includes the preferences of the agents. We hasten to add that this does not mean that the solutions are not clever or useful; they are. However, to prove that agents will actually follow a protocol, one needs a game-theoretic definition of the SFE problem. It turns out that the game theoretic analysis provides a slightly different perspective on (e.g.,) SFE; the paranoias of game theorists are more extreme than the traditional paranoias of cryptographers

in some respects and less so in others. The difference between the two demands a more complete discussion than we have space for, and we discuss the issue in more depth in a companion paper.

In this paper we do not speak about cryptography *per se*, but rather about a general framework within which to think about cryptography and related phenomena. The framework is called *non-cooperative computing*, or NCC for short. The term was introduced by Shoham and Tennenholtz in [4], who adopt a narrower setting. The NCC framework of S&T is however too limited to account for (e.g.,) cryptography, and the goal of this paper is to extend it so it does.

We give the formal definitions in the next section, but let us describe the NCC framework intuitively. The setting includes n agents and an n-ary function f, such that agent i holds input i. Broadly speaking, all the agents want to compute f correctly, but in fact each agent has several independent considerations. In this article we take agent i's utility function to depend on the following factors:

Correctness: i wishes to compute the function correctly.

Exclusivity: i wishes that the other agents do not compute the function correctly.

Privacy: i wishes that the other agents do not discover i's private input.

Voyeurism: i wishes to discover the private inputs of the other agents.

Of course, these considerations are often conflicting. They certainly conflict across agents – one agent's privacy conflicts with another agent's voyeurism. But they also conflict within a given agent – the wish to compute the function may propel the agent to disclose his private input, but his privacy concerns may prevent it. So the question is how to amalgamate these different considerations into one coherent preference function.

In this paper we consider lexicographic orderings. In the extended abstract, we analyze all 24 possible orderings of these considerations, while in the full paper we consider all possible orderings on all subsets of the considerations. In each case we ask for which functions f there exists a mechanism in the sense of mechanism design [3], such that in the game induced by the mechanism, it is a Bayes-Nash equilibrium for the agents to disclose their true inputs. Of course, to do that we must be explicit about the probability distribution from which the agents' inputs are drawn.

This is a good point at which to make clear the connection between our setting and the restricted NCC setting of S&T:

- S&T consider only *correctness* and *exclusivity* (and, in particular, only the ordering in which *correctness* precedes *exclusivity*).
- S&T consider both the case in which the inputs of the agents are independently distributed and the case in which they are correlated.

- S&T consider also a version of the setting in which agents are willing to mis-compute the function with a small probability, and another version in which agents can be offered money, in addition to their inherent informational incentives.

We not only consider *privacy* and *voyeurism* in addition to *correctness* and *exclusivity*, but also consider all 24 possible orderings among them (mercifully, in the Boolean case which we investigate they collapse to four equivalence classes), maintaining the property that all agents have the same ordering over the considerations. However, in this paper we do not investigate the case of correlated values, nor the probabilistic and monetary extensions. We leave those to future work.

There is one additional sense in which our treatment is more general. Consider for example the consideration of *correctness*, and three possible outcomes: in the first the agent believes the correct value with probability .6, in the second with probability .99, and in the third with probability 1. Holding all other considerations equal, how should the agent rank these outcomes? Clearly the third is preferred to the others, but what about those two? Here we have two versions; in one, the first two are equally desirable (in other words, any belief less than certainty is of no value), and in the other the second is preferred to the first. We call those two settings the *full information gain* setting and the *partial information gain* setting, respectively.

This means that rather than 24 cases we need to investigate, we have 48. But again, luck is on our side, and we will be able to investigate a small number of equivalence classes among these cases.

In the next section we give the precise definitions, and in the following sections we summarize our results. Several of the insights into our results are derived from the results obtained by S&T; we will try to indicate clearly when that is the case.

2 Formulation

2.1 Formal problem definition

As in NCC, let $N = \{1, 2, \ldots, n\}$ be a set of agents, and consider also a non-strategic center which will execute the protocol. We assume that each agent $1 \ldots n$ has a private and authenticated channel between itself and the center. Each agent has an input V_i drawn from the set B_i. We will use v_i (as shorthand for $V_i = v_i$) to represent a particular (but unspecified) value for V_i. The vector $v = (v_1, \ldots, v_n)$ consists of the types of all agents, while $v_{-i} = (v_1, \ldots, v_{i-1}, v_{i+1}, \ldots, v_n)$ is this same vector without the type of agent i ($v_{-i,j}$ simply extends this to removing two types). $P(V)$ is the joint probability over all players' inputs, which induces a $P_i(V_i)$ for each agent. Each agent knows his own type, but does not know the types of the other agents. Instead, the prior $P(V)$ (which we assume has full support – that is, $\forall v\ P(v) > 0$) is common knowledge among the agents and known by the mechanism designer. We further assume that the agent types are independent.

The commonly-known function that the agents are trying to compute is denoted by $f : B_1 \times \ldots \times B_N \rightarrow B_0$. Though the setting makes sense in the case of an arbitrary field, we restrict ourselves in this work to the case of Boolean functions over Boolean inputs ($B = B_i = \{0, 1\}$). We assume that all agents are *relevant* to the function in that they have at least some chance of affecting the outcome. Formally, this means that, for each agent i, $\exists v_i, v_{-i}\ f(v_i, v_{-i}) \neq f(\neg v_i, v_{-i})$.

2.2 The mechanism design problem

In general, a mechanism is a protocol specifying both the space of legal messages that the individual agents can send to the center and, based on these messages, what the center will return to each agent. As mechanism designers, our goal is for all agents, for all possible input values, to believe the correct value of the function at the end of the protocol. Since dominant strategy implementation is not feasible in our setting, we are looking for a Bayes-Nash implementation.

A standard mechanism is a mapping from actions to outcomes. The setting of NCC is somewhat different from the standard mechanism design setting, however. In the case of NCC, an outcome consists of a belief state for each agent, where a belief state is defined by a probability distribution over the output of the function and the inputs of the other agents. Instead of mapping from actions to outcomes, the mechanism in the NCC setting instead gives a signal to each player, who interprets it according to his *belief strategy*. Thus, a mechanism cannot enforce outcomes: it can only control the information embedded in its signal to each player. As we shall see, this will be sufficient for our purposes. A player's preferences over his and others' belief states are defined with respect to the correct inputs to and outputs of the function, as determined by the private types of the other players.

A priori, one could imagine arbitrarily complicated mechanisms in which the agents and the center iterate through many rounds of messages before converging on a result. However, following Shoham and Tennenholtz, we note that an extended revelation principle (extended from, e.g., [3]) allows us wlog to restrict our attention to mechanisms in which the agents truthfully declare an input to the center and accept the result returned to them.

Theorem 1 (Extended Revelation Principle) *If there exists a protocol for the center and the agents in which each agent computes the correct value of the function, then there exists a truthful, direct protocol in which each agent accepts the center's output and thereby computes the correct value of the function.*

Formally, a mechanism is a tuple $(S_1, \ldots, S_n, g_1, \ldots, g_n)$, consisting of, for each agent i, a strategy space S_i and a function g_i that determines the output returned to the agent. A strategy of agent i consists of the following tuple of functions: $(s_i : B \rightarrow \triangle B, b_i^f : B \times B \rightarrow \triangle B, b_i^1 : B \times B \rightarrow \triangle B, ..., b_i^n : B \times B \rightarrow \triangle B)$. The first maps an agent i's true type to a distribution over its declared type (which we will sometimes refer to as $\hat{v}_i$). The second maps i's true type v_i

and the center's response $g_i(\hat{v})$ to i's beliefs about the output of the function f. The remaining functions map i's type and the center's response to i's beliefs about each agent j's private input. We shall henceforth refer to the tuple of belief functions, which together map to a complete belief state of agent i, as b_i, the agent's belief strategy. Agents may have other higher-order beliefs, but we can neglect these since they are not relevant to any agent's preferences.

The set of outcomes O is the set of all possible vectors of belief states (one for each agent) over the input and output values; that is, $O = (\triangle B \times \triangle B^n)^n$. We wish to implement the social choice function W which always selects an outcome in which, for all agents i, $Pr(b_i^f(v_i, g_i(\hat{v})) = f(v)) = 1$. That is, in our desired outcome, each agent always computes the correct value of the function. In this paper, we restrict the range of $g_i : B^n \rightarrow B$ so that it returns to agent i a bit (to represent a possible output of the function) for each set of declared values $\hat{v}$. Since we wish every agent to always compute correctly, we can restrict the center's protocol to computing and returning the function $f(\hat{v})$ to each player (i.e. $g_i(\hat{v}) = f(\hat{v})$).

Each agent's preferences are over all of the beliefs that make up the outcome and over the true inputs, which determine the correctness of the beliefs. An agent desires that its own beliefs over the inputs and the output of the function are correct, and that those of the other agents are not. We now give a more formal definition of the incentives of each agent, first for the *full information gain* setting.

Correctness: i wishes that $Pr(b_i^f(v_i, g_i(\hat{v})) = f(v)) = 1$.

Exclusivity: For each $j \neq i$, i wishes that $Pr(b_j^f(v_j, g_j(\hat{v})) = f(v)) \neq 1$.

Privacy: For each $j \neq i$, i wishes that $Pr(b_j^i(v_j, g_j(\hat{v})) = v_i) \neq 1$.

Voyeurism: For each $j \neq i$, i wishes that $Pr(b_i^j(v_i, g_i(\hat{v})) = v_j) = 1$.

In the *partial information gain* setting, agent valuations depend on more than whether or not a probability is equal to 1. Instead, agents attempt to maximize the entropy function, which for a distribution $Pr(X)$ over a Boolean variable X is defined as: $H(X) = -Pr(X = 0) \cdot log_2 Pr(X = 0) - Pr(X = 1) \cdot log_2 Pr(X = 1)$.

Because of the way in which we can reduce the space of mechanisms that we need to consider, we can restate our goal as follows. In this paper we characterize, for each possible ordering on the four incentives listed above, the set of functions f for which it is a Bayes-Nash equilibrium for each agent i to use a strategy $(s_i(v_i) = v_i, b_i^f(v_i, f(v)) = f(v), ...)$ – that is, always telling the truth and believing the output of the mechanism.

3 Full information gain setting

In this section we consider the *full information gain* setting, in which we assume that agents are only concerned with what they and the other agents know with certainty, as opposed to what

they can know with some probability. We now characterize the set of functions that are NCC for each of 24 possible orderings of the incentives, which can be broken into four cases.

Before we begin, we review two definitions and a theorem from S&T [4] that will play an important role in our impossibility results. We say that f is *(locally) dominated* if there exists a type for some agent which determines the output of the function. Formally, the condition is that there exists an i and v_i such that $\forall v_{-i}, v'_{-i},\ f(v_i, v_{-i}) = f(v_i, v'_{-i})$. We say that a function f is *reversible* if there exists an agent i such that $\forall v_{-i},\ f(V_i = 0, v_{-i}) \neq f(V_i = 1, v_{-i})$, which means that for either input, agent i knows what the value of function would have been if he had submitted the other input.

Theorem 2 (Shoham and Tennenholtz) *When agents value* correctness *over* exclusivity*, and value no other consideration, a function is NCC if and only if it is non-reversible and non-dominated.*

We can restrict the class of functions to consider in the current setting by noting that any function which is not NCC in the S&T sense is not NCC for any ordering of the four incentives.

Theorem 3 *Any function that is reversible or dominated is not NCC.*

3.1 Exclusivity and correctness

We can tackle half of the orderings at once by considering the case in which all agents rank *exclusivity* over *correctness*. Not surprisingly, all is lost in this case.

Theorem 4 *If* exclusivity *is ranked over* correctness*, then no function is NCC.*

On the other hand, we find that the converse of Theorem 3 holds when *correctness* is ranked above all other factors.

Theorem 5 *If* correctness *is ranked over all other factors, then a function is NCC if and only if it is non-reversible and non-dominated.*

3.2 Privacy over correctness

We are now down to six cases, in which *correctness* must be ranked second or third, and *exclusivity* must be ranked below it. For the four of these cases in which *privacy* is ranked over *correctness*, the key concept is what we call a *privacy violation*, which occurs when an agent has an input for which there is a possible output that would allow the agent to determine another agent's input with certainty. Formally, we say that a *privacy violation* for agent i by agent j occurs whenever $\exists v_j, x, y, \forall v_{-j}\ (f(v_j, v_{-j}) = x) \Rightarrow (V_i = y)$.

Theorem 6 *If* privacy *is ranked over* correctness*, and both are ranked over* exclusivity*, then a function is NCC if and only if it is non-reversible, non-dominated, and has no privacy violations.*

It is interesting to note the relationship between privacy violations and what we call *conditional (local) domination*. We say that agent i *conditionally dominates* f given agent j if $\exists v_i, v_j, x\ (\forall v_{-i,j},\ f(v_i, v_j, v_{-i,j}) = x) \wedge (\exists v_{-i,j}\ f(\neg v_i, v_j, v_{-i,j}) \neq x)$. Using the terminology we defined earlier, conditional domination occurs when agent j can submit an input v_j such that agent i both dominates and is relevant to the output of the conditional function $f_{-j}(v_{-j}) = f(v_j, v_{-j})$.

Lemma 7 *: There exists a privacy violation for agent i by agent j if and only if agent i conditionally dominates f given j.*

3.3 Voyeurism first, correctness second

The final two cases to consider are those in which the first two considerations are *voyeurism* and *correctness*, in that order. If there exists an agent j who can obtain a greater amount of *voyeurism*, on expectation, from one of his possible inputs, then he will always choose to declare this input. Thus, a necessary condition for the function to be NCC is that the expected *voyeurism* be equal for $V_j = 0$ and $V_j = 1$. If this is the case, then *correctness* becomes paramount, and we again have the classic NCC condition.

Formally, define a new indicator function $violate(i, j, v_j, x)$ to be 1 if a privacy violation occurs for agent i by agent j, and v_j and x satisfy the condition for the violation to occur, and 0 otherwise. Now, we can formally give the condition for a *voyeurism tie*.

Theorem 8 *If all agents rank* voyeurism *first and* correctness *second, then, given a prior P(V), a function is NCC if and only if it is non-reversible and non-dominated and the following condition for a* voyeurism tie *holds for each agent j:*

$$\sum_{i \neq j} \sum_{v_{-j}} P(v_{-j}) \cdot violate\Big(i, j, V_j = 1, f(V_j = 1, v_{-j})\Big) = \sum_{i \neq j} \sum_{v_{-j}} P(v_{-j}) \cdot violate\Big(i, j, V_j = 0, f(V_j = 0, v_{-j})\Big)$$

For the common prior $P(0) = \frac{1}{2}$, an example of a function for which there is a voyeurism tie in the presence of privacy violations is the *unanimity* function, in which $f(v) = 1$ if and only if the inputs of all agents are identical.

Finally, note that the space of functions which are NCC in these two cases is a superset of the functions which are NCC in the cases of the previous subsection (*privacy* over *correctness*), since a complete lack of privacy violations trivially induces a voyeurism tie.

4 Partial information gain setting

Now we consider the *partial information gain* setting, in which agents value increased information about a factor. For this setting, we see that the results are unchanged for many of the possible lexicographic orderings, but are different for several interesting cases.

4.1 Unchanged results

The first three theorems from the *full information gain* setting carry over exactly to this setting.

Theorem 9 *In the* partial information gain *setting, any function that is reversible or dominated is not NCC.*

Theorem 10 *In the* partial information gain *setting, if agents rank* exclusivity *over all other factors, then no function is NCC.*

Theorem 11 *In the* partial information gain *setting, if agents rank* correctness *over all other factors, then a function is NCC if and only if it is non-reversible and non-dominated.*

4.2 Privacy over correctness

For the cases in which *privacy* is ranked above *correctness*, the condition for non-cooperative computability becomes more stringent, and it now depends on the form of the prior.

First, we need to update the definition of a privacy violation, because it now occurs whenever an agent's posterior distribution over another agent's input differs at all from the prior distribution. We say that a *partial privacy violation* of agent i by agent j occurs if $\exists v_i, x,\ Pr(V_j|v_i, f(v) = x) \neq P(V_j)$.

Theorem 12 *In the* partial information gain *setting, if agents rank* privacy *over* correctness*, then, given a prior $P(V)$, a function is NCC if and only if it is non-reversible and non-dominated and there are no partial privacy violations.*

The absence of partial privacy violations can also be formulated by the following condition, which, in words, requires that no pair of inputs provide more information about the output than any other pair.

Lemma 13 *A function has no partial privacy violations if and only if satisfies the following condition:*

$$\exists c,\ \forall i, j, v_i, v_j,\ Pr(f(v) = 0|v_i, v_j) = c$$

A (relatively) simple function that satisfies this condition is: $f(v) = \mathit{parity}(v_1, v_2, v_3) \wedge \mathit{parity}(v_4, v_5, v_6)$, with a common prior of $P(0) = \frac{1}{2}$. There also exist privacy-preserving functions that treat each agent's input symmetrically. One example, for $N = 7$ and the common prior $P(0) = \frac{1}{2}$, is the function $f(v)$ that returns 1 if and only if the number of agents i for which $v_i = 0$ is 1, 2, 4, or 5.

4.3 Voyeurism first, correctness second

The last two cases to consider are those in which agents rank *voyeurism* first and *correctness* second, leaving *privacy* as their third or fourth priority.

In order to calculate the expected entropy that agent j has agent i's input after learning the output of the function, we can use the following expression: $Pr(v_i|f(v) = x, v_j) = \frac{Pr(f(v)=x|v_i,v_j)\cdot P(v_i)}{Pr(f(v)=x|v_j)}$.

For the desired equilibrium to hold, it must be the case that for each agent i the expected partial *voyeurism* is the same for both possible values of V_i.

Theorem 14 *In the* partial information gain *setting, if agents rank* voyeurism *first and* correctness *second, then, given a prior P(V), a function is NCC if and only if it is non-reversible and non-dominated and the following condition holds for each agent j:*

$$\sum_{i \neq j} E_x[H(V_i|V_j = 1, f(v) = x)] \neq \sum_{i \neq j} E_x[H(V_i|V_j = 0, f(v) = x)]$$

Note that, for the common prior $P(0) = \frac{1}{2}$, the *unanimity* function still induces a voyeurism tie, as it did for the same two orderings of the *full information gain* setting.

5 Conclusion

In this paper, we have considered a class of incentive structures for agents and a class of mechanisms, and characterized the sets of functions which are computable by agents which are similarly self-interested. A summary of our results lends itself to a decision tree, as shown in Figure 1.

We view these results as laying the groundwork for a consideration of a wide variety of both theoretical concerns and practical problems. In the introduction, we discussed the cryptographic problem of secure function evaluation. Determining whether these cryptographic protocols will lead to successful computation requires considering not only deviations from the protocol given agent inputs (which is the extent of the analysis in most papers in this field), but also whether the protocol is incentive compatible. Using the impossibility results stated above, we can focus our efforts on designing protocols for functions which are non-cooperatively computable.

In addition, we can extend our formulation along several dimensions. For example, if we allow the mechanism to return to an agent the inputs of other agents, in addition to the output of the function, then *voyeurism* no longer prevents a function from being NCC. Intuitively, the mechanism will expose inputs in a way that always creates a voyeurism tie. While similar solutions cannot overcome *privacy* or *exclusivity*, other extensions, including correlation between agent inputs and the possibility of monetary payments, further expand the space of functions that are NCC. Finally, the analysis and types of results we obtained are not limited to Boolean functions and lexicographic utility functions, and we regard extending this analysis to more general fields as a promising line of future research.

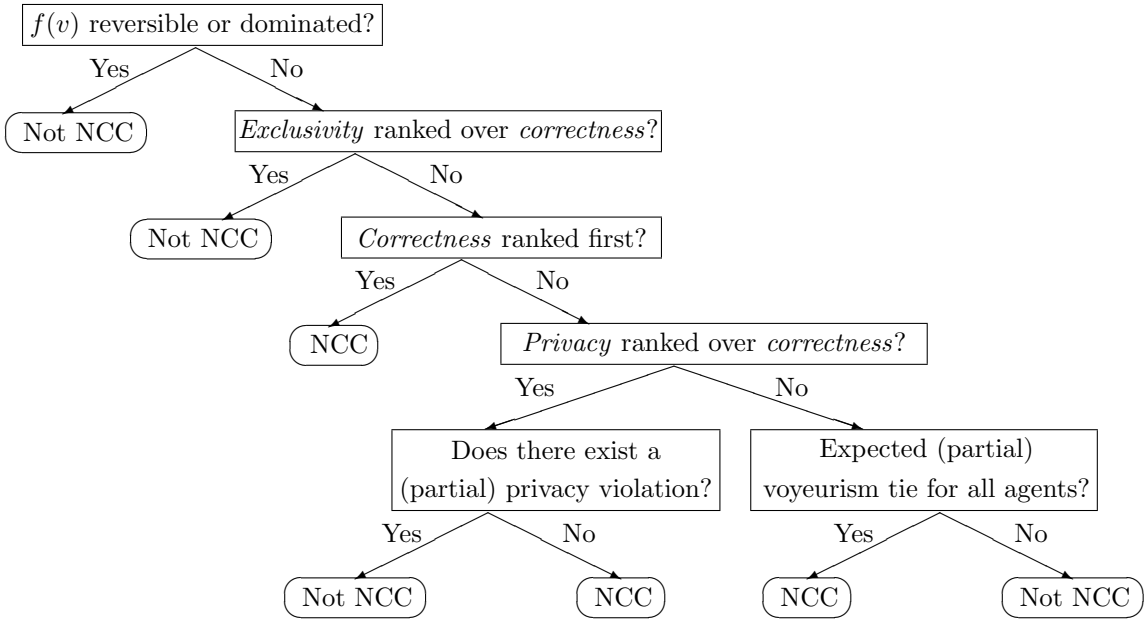

Figure 1: A decision tree which summarizes the conditions for a function and prior to be NCC. The conditions are the same for the two settings we consider, except for the bottom two decision boxes, in which "(partial)" refers to the updated conditions for the *partial information gain* setting.

References

[1] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. STOC'88.

[2] Shafi Goldwasser. Multi party computations: past and present. Proceedings of the sixteenth annual ACM symposium on Principles of distributed computing, 1989. ACM.

[3] A. Mas-Collel, W. Whinston, and J. Green. Microeconomic Theory. 1995.

[4] Yoav Shoham and Moshe Tennenholtz. Non-Cooperative Evaluation of Logical Formulas: The Propositional Case. Under review. 2003.