Computational Extensive-Form Games

Joseph Y. Halpern Rafael Pass Lior Seeman
Computer Science Dept.
Cornell University
Ithaca, NY
E-mail: halpern|rafael|lseeman@cs.cornell.edu

Abstract

We define solution concepts appropriate for computationally bounded players playing a fixed finite game. To do so, we need to define what it means for a *computational game*, which is a sequence of games that get larger in some appropriate sense, to represent a single finite underlying extensive-form game. Roughly speaking, we require all the games in the sequence to have essentially the same structure as the underlying game, except that two histories that are indistinguishable (i.e., in the same information set) in the underlying game may correspond to histories that are only computationally indistinguishable in the computational game. We define a computational version of both Nash equilibrium and sequential equilibrium for computational games, and show that every Nash (resp., sequential) equilibrium in the underlying game corresponds to a computational Nash (resp., sequential) equilibrium in the computational game. One advantage of our approach is that if a cryptographic protocol represents an abstract game, then we can analyze its strategic behavior in the abstract game, and thus separate the cryptographic analysis of the protocol from the strategic analysis. Finally, we use our approach to study the power of having memory in a TM. Specifically, we show that there is a gap between what can be done with stateful strategies (ones that make use of memory) and what can be done with stateless strategies.

1 Introduction

Game-theoretic models assume that the player are completely rational. This is typically interpreted as saying that payers act optimally given (their beliefs about) other players' behavior. However, as was first pointed out by Simon [17], acting optimally may be hard. Thus, there has been a great deal of interest in capturing *bounded rationality*, and finding solution concepts appropriate for resource-bounded players.

One explanation of bounded rationality is that players have limits on their computational power. There have been two dominant approaches for modeling such limitations. One approach, initiated by Rubinstein [16], is to view the players as choosing an algorithm, and to have a player's utility depend in part on the computational resources used by the algorithm. For example, if we model the algorithm by a finite automaton, the utility could depend on the size (number of states) of the automaton used [1; 16]; more generally, as suggested by Halpern and Pass [7], players could choose a Turing machine (TM) and the utility could depend on both the TM chosen and its input. A second approach, initiated by Neyman [14], is to restrict players to choosing an algorithm in some restricted set that is meant to capture their computational limitations. For example, Neyman [14] viewed players as finite automata, and Urbano and Vila [18] and Dodis, Halevi and Rabin [3] modeled players as polynomial-time TMs.

While there has been a great deal of work, especially recently, on solving game-theoretic problems using polynomial-time TMs, there has not really been a careful study of the solution concepts

appropriate for such resource-bounded players. What does it mean, for example, to say that a fixed finite game played by polynomial-time players has a Nash equilibrium (NE)? To get a sense of the problems, note that to talk about polynomial time, we need to have a set of inputs that can grow as a function of n. When considering polynomial-time players in repeated games, we can consider longer and longer repetitions of the game (this was done, for example, in [2, 10]), but how do we proceed if we want to talk about equilibria for polynomial-time players in a fixed finite game? Another complication is that NE involves all players making a best response. But if we restrict to polynomial-time players, there may not be a best response, especially for the kinds of cryptographic problems that we would like to consider. For every polynomial-time TM, there may be another TM that does a little better by spending a little longer trying to do decryption. (See [8] for an example of this phenomenon.)

As a first step to capturing these notions, after reviewing some relevant background in game theory and cryptography in Section 2, in Section 3 we define what it means for a sequence $\mathcal{G} = (G_1, G_2, \ldots)$ of games to represent a single game G. Intuitively, all the games in the sequence $\mathcal{G}$ represent G, but might use increasingly longer strings to represent actions in G (for example, might use an encryption of the action using increasingly longer security parameters). Thus, in a sense, the games G_1, G_2, G_3, ... grow larger, so serve as the input to a polynomial-time TM. To make sense of the idea of the games in the sequence all representing G, we define a mapping from histories in the games G_n to histories in G, and impose what we argue are reasonable conditions on the mapping.[1]

Interestingly, our conditions do not force the same information structure on both G and $\mathcal{G}$. While two histories in the same information set in G_n must map to two histories in the same information set in the underlying game G, it may also be the case that two histories in different information sets in G_n are mapped to the same information set in G. To understand why we want to allow this, suppose that in G_n, there are histories h_1 and h_2 where a bit 0 in encrypted using two different keys by agent 1. Agent 2 can distinguish h_1 and h_2, because the encryptions are two different strings; thus, h_1 and h_2 are in different information sets for agent 2. And they are both in a different information set from h_3, where a bit 1 is encrypted. Nevertheless, both h_1 and h_2 are mapped to the same history h in G, where an unencrypted bit 0 is put in an envelope, while h_3 is mapped to a history h', where an unencrypted bit 1 is put in an envelope, which is in the same information set as h. (See the example in Section 3.2 for more intuition.)

Although agent 2 can distinguish histories h_1, h_2, and h_3 above, at a computational level, she cannot tell them apart. The encodings just look like random strings to her. There is a sense in which she, as a polynomial-time player, does not understand the "meaning" of these histories (although a computationally unbounded player could break the encryption and tell them apart). We make this intuition precise, showing that our requirements force all histories that map to the same information set in G to be computationally indistinguishable (by a polynomial-time agent), even if they are in different information sets in $\mathcal{G}$.

Once we have defined our model of computational games, we can consider solutions concepts. We focus on two solution concepts here, Nash equilibrium and sequential equilibrium; in Section 4, we define computational analogues of both. We then show that if a strategy profile is a Nash (resp., sequential) equilibrium in the underlying game G, then there is a corresponding strategy profile of polynomial time TMs that is a computational Nash (resp., sequential) equilibrium in $\mathcal{G}$.

It is notoriously problematic to define sequentially rational solution concepts in cryptographic protocols. For example, Gradwohl, Livne, and Rosen [6] provide a general discussion of the issue, and give a partial solution in terms of avoiding what they call "empty threats", which applies only to two-player games of perfect information, and discuss possible extensions. Our notion of

[1]The idea of describing a solution concept that depends on a security parameter goes back to Dodis, Halevi and Rabin [3] Hubáček and Park [11] also consider a mapping between histories in a computational game and histories in an abstract game, although they do not consider the questions in the same generality that we do here.

computational sequential equilibrium, which is quite different in spirit from the solution concepts of Gradwohl, Livne, and Rosen (and arguably conceptually much simpler and much closer in spirit to the standard game-theoretic definition) applies to arbitrary finite games; it thus may give further insight into issues as incredible threats. We show in Section 6 that our approach leads to an arguably much simpler and more natural analysis of a protocol for implementing a correlated equilibrium without a mediator.

Our work also gives insight into one other issue: the power of state in a TM. In our model, we assume that TMs have state, that is, a separate memory tape in which they can store information (such as the randomness used in earlier rounds), which can then be used in later rounds. For example, a TM can reconstruct the encryption key used in an earlier round by looking at the randomness used in creating it. Since storing information may be expensive, a desirable property for a protocol is that it be stateless (where a *stateless TM* is one whose next action can depend only on the history of play).

Stateful TMs seem necessary in order to implement mixed strategies, that is, distributions over pure (deterministic) strategies. A TM plays a mixed strategy by initially tossing some coins, whose outcome determines which pure strategy it plays. It must be able to access the outcome of the initial coin tosses so that it can know what strategy to use in later rounds. On the other hand, a stateless TM can implement *behavioral strategies* (which are functions from information sets to distribution over actions) but, intuitively, cannot implement mixed strategies.

Kuhn [13] proved that for every mixed strategy profile in a finite extensive-form game with *perfect recall* (where agents recall all the actions that they have performed and all the information sets that they have gone through), there is a behavioral strategy profile in the game that is equivalent in the sense of inducing the same distribution over terminal histories. This is not necessarily the case in games of imperfect recall [19]. There is an analogy between perfect vs. imperfect recall and mixed vs. behavioral stratgies on the one hand, and polynomial-time vs. unrestricted computation and stateful vs. stateless TMs on the other. If we restrict to polynomial-time players, then in computational games, not every strategy profile with stateful TMs is equivalent to a profile with stateless TMs, at least under standard cryptographic assumptions; however, it is not hard to show that for computationally unrestricted players, stateful and stateless TMs are equivalent. For example, a stateful TM can use a random key to commit to a bit and later always open the commitment correctly. If there exists a stateless TM that implements the same distribution, it must be the case that it is able to break the commitment, which a polynomial-time player cannot do. On the other hand, with unrestricted computation, a stateless TM can simulate a stateful TM by resampling a consistent random string.

In Section 5, we actually prove an even stronger result. As we said above, if $\mathcal{G}$ represents a game G, then to every NE in G, there is a corresponding computational NE in $\mathcal{G}$. This computational NE is a mixed strategy, which we model as a (stateful) TM. We show that, under a standard cryptographic assumption, namely, that exponentially hard one-way permutations exist, there are computational games with perfect recall for which there is no computational NE using stateless TMs. The key step in the proof is to construct a game where, by using the exponentially hard permutation, given a TM M_2 for the second player, a stateless TM can deviate by choosing an encryption key that is just long enough to "fool" M_2, while making sure it is short enough so it can itself reconstruct the state later. On the other hand, for any stateless TM M_1 for the first player, the second player's TM can just simulate M_1 up to the point where it reveals the commitment (since the history is M_1's input); thus, M_2 learns M_1's output, and can use it to break the encryption.

This distinction between stateful and stateless TMs has already arisen in other contexts. Borgs et al. [2] showed that, in general, we cannot compute a NE in a repeated game in polynomial time; in [10], we showed that, under standard cryptographic assumptions, we could compute a NE (indeed, even a sequential equilibrium [9]) in polynomial time. The reason that we were able to

obtain our positive result was that (1) we restricted to only polynomial-time deviations (as we do in this paper as well) and (2) we assumed stateful TMs, while Borgs et al. assumed stateless TMs. These results show that there are some subtle issues that must be addressed when modeling polynomial-time players.

The result on stateless TMs is an example of the subtleties that arise when trying to analyze cryptographic protocols from a game-theoretic perspective. Using our approach, we can separate the game-theoretic analysis from the cryptographic analysis. We can view the sequence $\mathcal{G}$ as an implementation of an abstract game G. Under this view, the relationship between $\mathcal{G}$ and G is similar in spirit to the relationship between ideal and real worlds often used in describing cryptographic protocols. We can view the ideal protocol as an abstract game G and the sequence $\mathcal{G}$ as implementation of it, using increasing security parameters. Given this view, we can first prove that a protocol is a good implementation of an abstract game, and then analyze the strategic aspects in that simple abstract game. For example, to show a prescribed cryptographic protocol is a Nash (resp., sequential) equilibrium, we can first show it represents an abstract ideal game; it then suffices to show that the protocol corresponds to a strategy profile that is a Nash (resp., sequential) equilibrium in the much simpler underlying game.

2 Preliminaries

2.1 Extensive-form games

We begin by reviewing the formal definition of an extensive-form game [15]. A finite extensive-form game G is a tuple $([c], H, P, \vec{u})$, where

- $[c] = \{1, \ldots, c\}$ is the set of players in the game;
- H is a set of history sequences that satisfies the following two properties:
  - the empty sequence is a member of H.
  - if $\langle a_1, \ldots, a_K \rangle \in H$ and $L < K$ then $\langle a_1, \ldots, a_L \rangle \in H$. The elements of a history h are called *actions*.

  A history $\langle a_1, \ldots, a^K \rangle \in H$ is *terminal* if there is no a such that $\langle a^1, \ldots, a^K, a \rangle \in H$. The set of actions available after a nonterminal history h is denoted $A(h) = \{a : h \cdot a) \in H\}$ (where $h \cdots a$ is the result of concatenating a to the end of h.[2] Let H^T denote the set of terminal histories, let H^{NT} denote $H \setminus H^T$, and let H^i denote the histories after which player i plays.
- A function $P : H \setminus H^T \to [c]$. $P(h)$ specifies the player that moves at history h.
- $\vec{u} : H_T \to \mathbb{R}^c$ specifies for each terminal history the utility of the players at that history ($u_i(h)$ is the utility of player i at terminal history h).
- For each player $i \in [c]$, a partition $\mathcal{I}_i$ of H^i with the property that $A(h) = A(h')$ whenever h and h' are in the same member of the partition. For $I \in \mathcal{I}_i$ we denote by $A(I)$ the set $A(h)$ for $h \in I$ (recall that $A(h) = A(h')$ if h and h' are two histories in I). We assume without loss of generality that if $I \neq I'$, then $A(I)$ and $A(I')$ are disjoint (we can always rename actions to ensure that this is the case). We call $\mathcal{I}_i$ the *information partition* of player i; a set $I \in \mathcal{I}_i$ is an *information set* of player i; $\vec{\mathcal{I}} = (\mathcal{I}_1, \ldots, \mathcal{I}_c)$ is the *information partition structure* of the game. A game of *perfect information* is one where all the information sets are singletons.

[2] For technical convenience, we assume that $|A(h)| \geq 2$ for all histories h. If this is not the case, then that step of the game is not interesting, and can essentially be removed.

This model can capture situations in which players forget what they knew information structure is such that the players remember everything they knew in the past.

Definition 2.1. *Let $EXP_i(h)$ be the record of player i's* experience *in history h, that is, all the actions he plays and all the information sets he encounters in the history. A game has* perfect recall *if, for each player i, we have $EXP_i(h) = EXP_i(h')$ whenever the histories h and h' are in the same information set for player i.*

A deterministic strategy s for player i is a function from $\mathcal{I}_i$ to actions, where for $I \in \mathcal{I}_i$, we require that $s(I) \in A(I)$. We also consider randomized strategies. In the literature, two types of randomized strategies have been considered:

- *mixed* strategies: a mixed strategy σ^m for player i is a probability distribution over deterministic strategies.
- *behavioral* strategies: a behavioral strategy σ^b for player i maps I_i to distributions over actions such that for all action a in the support of $\sigma(h)$, $a \in A(I_i)$.

A profile of strategies (mixed or behavioral) $\sigma = \{\sigma_1, \ldots, \sigma_c\}$ induces a distribution denoted ρ_σ on terminal histories. The expected value of player i given σ is then $\sum_{h \in H^T} \rho_\sigma(h) u_i(h)$. Kuhn [13] shows that for every mixed strategy profile for a player in a finite extensive-form game with perfect recall there is a behavioral strategy profile for the players in the game that induces the same distribution over terminal histories. This is not necessarily the case in games of imperfect recall (see, for example, [19]).

We use the standard notation $\vec{x}_{-i}$ to denote the vector $\vec{x}$ with its ith element removed and $(x', \vec{x}_{-i})$ to denote $\vec{x}$ with its ith element replaced by x'.

Definition 2.2 (Nash Equilibrium)**.** *$\vec{\sigma} = \{\sigma_1, \ldots, \sigma_c\}$ is an* ϵ-Nash equilibrium *(NE) of G if, for all players $i \in [c]$ and for all strategies σ' for player i,*

$$\sum_{h \in H^T} \rho_{\vec{\sigma}}(h) u_i(h) \geq \sum_{h \in H^T} \rho_{\sigma', \vec{\sigma}_{-i}}(h) u_i(h) - \epsilon.$$

We now recall the notion of*sequential equilibrium* [12]. A sequential equilibrium is a pair $(\vec{\sigma}, \mu)$ consisting of a strategy profile $\vec{\sigma}$ and a *belief system* μ, where μ associates with each information set I a probability $\mu(I)$ on the nodes in I. Intuitively, if I is an information set for player i, $\mu(I)$ describes i's beliefs about the likelihood of being in each of the nodes in I. Then $(\vec{\sigma}, \mu)$ is a sequential equilibrium if, for each player i and each information set I for player i, σ_i is a best response to $\vec{\sigma}_{-i}$ given i's beliefs $\mu(I)$. An equivalent definition that does not require beliefs and is more suitable for our setting is given by the following theorem:

Theorem 2.3. *[12, Proposition 6] Let G be an extensive-form game with perfect recall. There exists a belief system μ such that $(\vec{\sigma}, \mu)$ is a sequential equilibrium of G iff there exists a sequence of completely mixed strategy profiles $\vec{\sigma}^1, \vec{\sigma}^2, \ldots$ converging to $\vec{\sigma}$ and a sequence $\delta_1, \delta_2, \ldots$ of nonnegative real numbers converging to 0 such that, for each player i and each information set I for player i, $\vec{\sigma}_i^n$ is a δ_n-best response to $\vec{\sigma}_{-i}^n$ conditional on having reached I.*

2.2 Computational indistinguishability

For a probabilistic algorithm A and an infinite bitstring r, $A(x; r)$ denotes the output of A running on input x with randomness r; $A(x)$ denotes the distribution on outputs of A induced by considering $A(x; r)$, where r is chosen uniformly at random. A function $\epsilon : \mathbb{N} \to [0, 1]$ is *negligible* if, for every constant $c \in \mathbb{N}$, $\epsilon(k) < k^{-c}$ for sufficiently large k. We say that that ϵ is *noticeable* if it is not negligible.

Definition 2.4. *A* probability ensemble *is a sequence $X = \{X_n\}_{n\in\mathbb{N}}$ of probability distribution indexed by $\mathbb{N}$. (Typically, in an ensemble $X = \{X_n\}_{n\in\mathbb{N}}$, the support of X_n consists of strings of length n.)*

We now recall the definition of computational indistinguishability [5].

Definition 2.5. *Two probability ensembles $\{X_n\}_{n\in\mathbb{N}}, \{Y_n\}_{n\in\mathbb{N}}$ are* computationally indistinguishable *if, for all PPT TMs D, there exists a negligible function ϵ such that, for all $n \in \mathbb{N}$,*

$$|\Pr[D(1^n, X_n) = 1] - \Pr[D(1^n, Y_n) = 1]| \leq \epsilon(n).$$

To explain the Pr *in the last line, recall that X_n and Y_n are probability distributions. Although we write $D(1^n, X_n)$, D is a randomized algorithm, so what $D(1^n, X_n)$ returns depends on the outcome of random coin tosses. To be a little more formal, we should write $D(1^n, X_n, r)$, where r is an infinitely long random bit strong (of which D will only use a finite initial prefix). More formally, taking $\Pr_{X_n}$ to be the joint distribution over strings (x, r), where x is chosen according to X_n and r is chosen according to the uniform distribution on bit-strings, we want*

$$|\Pr_{X_n}[\{(x, r) : D(1^n, x, r) = 1\}] - \Pr_{Y_n}[\{(y, r) : D(1^n, y, r) = 1\}]| \leq \epsilon(n).$$

We similarly abuse notation elsewhere in writing Pr*.*

We often call a TM M that is supposed to distinguish between two probability ensembles a *distinguisher*. We say that it distinguishes two ensembles *with overwhelming probability* if it distinguishes them with probability greater than $1 - \epsilon(n)$ for some negligible function ϵ.

3 Computational Extensive-Form Games

3.1 Motivation and definitions

Consider the following two-player extensive-form game G: At the the empty history, player 1 secretly chooses one of two alternatives and puts her choice inside a sealed envelope. Player 2 then also chooses one of these two alternatives. Finally, player 1 can either open the envelope and reveal her choice or destroy the envelope. If she opens the envelope and she chose a different alternative than player 2, player 1 wins and gets a utility of 1; otherwise (i.e., if player 1 either chose the same alternative as player 2 or she destroyed the envelope) player 1 loses and gets a utility of -1. Player's 2's utility is the opposite of player 1's. The game tree for this game is given in Figure 1. Since player 2 acts without knowing 1's choice, the two histories where 1 made different choices are in the same information set of player 2.

Resource-bounded players can implement this game even without access to envelopes, using what is called a *commitment scheme*. A commitment scheme is a two-phase two-party protocol involving a sender (player 1 above) and a receiver (player 2). The sender sends the receiver a message in the first phase that commits him to a bit without giving the receiver any information about the bit (this is the computational analogue of putting the bit in an envelope). In the second phase, the sender "opens the envelope" by sending the receiver some information that allows the receiver to confirm what bit the sender committed to in the first phase.

Definition 3.1. *A* secure commitment scheme with perfect bindings *is a pair of PPT algorithms C and R such that:*

- *C takes as input a security parameter 1^k, a bit b, and a bitstring r, and outputs $C(1^k, b, r), C_2(1^k, b, r)$, where $C_1(1^k, b, r)$, called the* commitment string*, is a k-bit string, and $C_2(1^k, b, r)$, called the* commitment key*, is a $(k-1)$-bit string. We use $C(1^k, b)$ to denote the output distribution of algorithm $C(1^k, b, r)$ when r is chosen uniformly at random.*

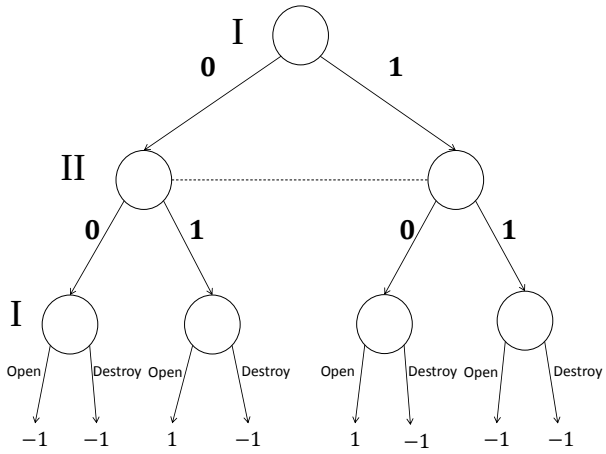

Figure 1: A game that can be represented by a computational game.

- *R is a deterministic algorithm that gets as input two strings c and s and outputs $o \in \{0, 1, f\}$.*
- *The ensemble $\{C_1(1^k, 0)\}_{k\in\mathbb{N}}$ is computationally indistinguishable from $\{C_1(1^k, 1)\}_{k\in\mathbb{N}}$.*
- *$R(C_1(1^k, b, r), (C_2(1^k, b, r)) = b$ and for all k and r; moreover, if $s \neq C_2(1^k, b, r)$, then $R(C_1(1^k, b, r)), s) \notin \{0, 1\}$.*

Cryptographers typically assume that secure commitment schemes with perfect bindings exist. (Their existence would follow from the existence of *one-way permutations*; see [4] for further discussion and formal definitions.)

We can model the simple game in Figure 1 using a commitment scheme. The point is that now we get, not one game, but a sequence of games, one for each choice of security parameter. Rather that putting a bit b in an envelope, player 1 sends $C(1^k, b)$. More precisely, he sends $C(1^k, b, r)$, for a string r chosen chosen uniformly at random. The fact that player 2 can't tell what bit player 1 sent is modeled by the indistinguishability of the ensembles $C_1(1^k, 0)$ and $C_1(1^K, 1)$. This is not information-theoretic indistinguishability; only a polynomial-time player cannot tell the ensembles apart. Thus, $C(1^k, 0, r)$ and $C(1^k, 1, r)$ are actually *not* in the same information set. We need to capture their computational indistinguishability another way.

Statements of computational difficulty typically say that there is no (possibly randomized) polynomial-time algorithm for solving a problem. To make sense of this, we need to consider, not just one input, but a sequence of inputs, getting progressively larger. Similarly, to make sense of computational games, we cannot consider a single game, but rather must consider a sequence of games that grow in size. The games in the sequence share the same basic structure. This means that, among other things, they involve the same set of players, playing in the same order, with corresponding utility functions. To make this precise, we first start with a more general notion, which we call a *computable uniform sequence of games.*

Definition 3.2. *A* computable uniform sequence $\mathcal{G} = \{G_1, G_2, \ldots\}$ of games *a sequence that satisfies the following conditions:*

- *All the games in the sequence involve the same set of players.*
- *Let H_n be the set of histories in G_n. There exists a polynomial p such that, for all nonterminal histories $h \in H_n^{NT}$, $A(h) \subseteq \{0, 1\}^{\leq p(n)}$.*[3] *In addition, there is a PPT algorithm that, on input 1^n and a history h, determines whether $h \in H_n$.*

[3] $\{0, 1\}^{\leq p(n)}$ denotes the language consisting of bitstrings of length at most $p(n)$.

- *There exists a polynomial-time computable function P' from $\bigcup_{n=1}^{\infty}(H_n^{NT})$ to $[c]$. The function P_n in game $G_n \in \mathcal{G}$ is then P' restricted to H_n^{NT}.*
- *For each player i, there exists a polynomial-time computable function $u_i : \bigcup_{n=1}^{\infty} H_n^T \to \mathbb{R}$ such that the utility function of player i in game G_n is u_i restricted to H_n^T.*

We sometimes call a computable uniform sequence of games a computational game.

Computable uniform sequences of games already suffice to allow us to talk about polynomial-time strategies. A strategy M for player i in a computable uniform sequence $\mathcal{G} = (G_1, G_2, \ldots)$ a probabilistic TM that takes takes as input a pair $(1^n, v)$, where v is a view for player i in G_n (discussed below), and outputs an action in $A(I)$. As we said in the introduction, we assume that the TMs have state (a tape on which information can be written). It suffices for our purposes in this paper that all that is written on this extra tape is the randomness that was used in earlier rounds; this suffices, for example, to reconstruct a secret key that was generated in the first round, so it can be used in later rounds. For ease of exposition in this paper, we take the TM's state to encode just this randomness. We say that such a TM is *stateful*. The *view* of a stateful TM M for player i in G_n is a tuple (v_I, r), where v_I is the representation of information set I and r contains the randomness that has been used thus far (so is nondecreasing from round to round). Of course, a stateless TM's view does not include the r component. M is a *polynomial-time TM for $\mathcal{G}$* if there exists a polynomial q such that, in G_n, M computes its next action using at most $q(n)$ steps.

We next define what it means for a uniform sequence $\mathcal{G} = (G_1, G_2, \ldots)$ of games to *represent* an underlying game G. For example, we want to discuss what it means for a sequence $\mathcal{G}$ to represent the game in Figure 1. Roughly speaking, we want all the games in $\mathcal{G}$ to have the same "structure" as G. We formalize this by requiring a surjective mapping f_n from histories in each game G_n in the sequence to histories in G. Note that f_n is not, in general, one-to-one. There may be many histories in G_n representing a single history in G. This can already be seen in our example; each of the histories in G_n where player 1 sends $C_1(1^n, b, r)$ get mapped to the history in G where player 1 puts 1 in an envelope. Moreover, although $C_1(1^n, 0, r_1)$ and $C_1(1^n, 0, r_2)$ get mapped to histories in the same information set in G, they are *not* in the same information set in G_n; an exponential-time player can break the encryption and tell that they correspond to different bits being put in the envelope. Thus, the mapping f_n does not completely preserve the information structure. We later discuss a sense in which the mapping does preserve the information structure. For now, we require that h and $f_n(h)$ have the same length. Of course, the utility associated with a terminal history h in G_n is the same as that associated with history $f_n(h)$ in G.

The first three conditions below capture the relatively straightforward structural requirements above. The final requirement imposes conditions on the players' strategies, and is somewhat more complicated. Informally, the fourth requirement is that there is a mapping $\mathcal{F}$ from strategies in G to a strategies in $\mathcal{G}$, where $\mathcal{F}(\sigma)$ "corresponds" to σ in some appropriate sense. But what should "correspond" mean? Let $\vec{M}$ be a strategy profile for $\mathcal{G}$. For each game $G_n \in \mathcal{G}$, $\vec{M}$ induces a distribution denoted $\psi_{\vec{M}}^{G_n}$ on the terminal histories in G_n. By applying f_n, we can push this forward to a distribution $\phi_{\vec{M}}^{G_n}$ on the terminal histories in G. A mixed strategy profile $\vec{\sigma}$ in G also induces a distribution on the terminal histories in G, denoted $\rho_{\vec{\sigma}}$.

Definition 3.3. *A strategy profile $\vec{\sigma}$* corresponds *to $\vec{M}$ if (1) $\{\phi_{\vec{M}}^{G_n}\}_{n\in\mathbb{N}}$ is* statistically close *to $\{\rho_{\vec{\sigma}}\}_{n\in\mathbb{N}}$: that is, if H^T are the terminal histories of G, then there exists a negligible function ϵ such that, for all n,*

$$\sum_{h\in H^T} |Pr_{\phi_{\vec{M}}^{G_n}}[h] - Pr_{\rho_{\vec{\sigma}}}[h]| \leq \epsilon(n).$$

So one requirement we will have is that, for all strategy profiles $\vec{\sigma}$ in G, $\vec{\sigma}$ corresponds to $(\mathcal{F}(\sigma_1), \ldots, \mathcal{F}(\sigma_n))$, which we abbreviate as $\mathcal{F}(\vec{\sigma})$. In addition, we require that the strategy profile $\mathcal{F}(\vec{\sigma})$ "knows" which underlying action it plays. We formalize this by requiring that, for strategy σ in the underlying game, there is a TM M^σ that, given view v for player i in $\mathcal{G}$, outputs the underlying action played by $\mathcal{F}(\sigma)$ given view v.

Finally, we require a partial converse to the correspondence requirement. It is clearly too much to expect a full converse. $\mathcal{G}$ has a richer structure than G; it allows for more ways for the players to coordinate than G. So we cannot expect very strategy profile in $\mathcal{G}$ to correspond to a strategy profile in G. Thus, we require only that strategies in a rather restricted class of strategy profiles in $\mathcal{G}$ correspond to a strategy in G: namely, ones where we start with a strategy of the form $\mathcal{F}(\vec{\sigma})$ (which, by assumption, corresponds to $\vec{\sigma}$), and allow one player to deviate. We must also use a weaker notion of correspondence here. For example, in the game in Figure 1, even if we start with a strategy of the form $\mathcal{F}(\vec{\sigma})$, the deviating strategy M_1' could be such that player 1 commits to 0 in G_n for n even, and commits to 1 in G_n for n odd. The strategy profile $(M_1', \mathcal{F}(\sigma_2))$ does not correspond to any strategy profile in G. Thus, the notion of correspondence that we consider in this case is that if i plays M_i' rather that $\mathcal{F}(\sigma_i)$, then there exists a sequence $\sigma_1', \sigma_2', \ldots$ of strategies in G, rather that a single strategy σ', and require only that the sequence $\{\phi^{G_n}_{(M_i', \mathcal{F}(\vec{\sigma}_{-i}))}\}_{n \in \mathbb{N}}$ be computationally indistinguishable from $\{\rho_{\vec{\sigma}}\}_{n \in \mathbb{N}}$, rather than being stastistically indistinguishable.

Definition 3.4. *A computable uniform sequence* $\mathcal{G} = \{G_1, G_2, \ldots\}$ represents *an underlying game* G *if the following conditions hold:*

UG1. G *and every game in* $\mathcal{G}$ *involve the same set of players.*

UG2. For each game $G_n \in \mathcal{G}$*, there exists a surjective mapping* f_n *from the histories in* G_n *to the histories in* G *such that (a)* $|h| = |f_n(h)|$*, (b) the same player moves in* h *and* $f_n(h)$*, (c) if* h' *is a subhistory of* h*, then* $f_n(h')$ *is a subhistory of* $f_n(h)$*, and (d) if* h *and* h' *are in the same information set in* G_n*, then* $f_n(h)$ *and* $f_n(h')$ *are in the same information set in* G*. (Note that it follows from these conditions that if* h *is a terminal history of* G_n*, then* $f_n(h)$ *must be a terminal history of* G*.) For* $h \in H$ *(a history of* G*), let* $LA(h)$ *denote the last action played in* h*. We additionally require that if* h *and* h' *are in the same information set in* G_n*, then for any* a *such that* $h||a \in H_n$*,* $LA(f_n(h||a)) = LA(f_n(h'||a))$ *(where* $||$ *is the concatenation operator).*

UG3. If h *is a terminal history of* G_n*, then the utility of each player* i *is the same in* h *and* $f_n(h)$*.*

UG4. There is a mapping $\mathcal{F}$ *from strategies in* G *to strategies in* $\mathcal{G}$ *such that*

- *(a) for all strategy profiles* $\vec{\sigma}$ *in* G*,* $\vec{\sigma}$ *corresponds to* $\mathcal{F}(\vec{\sigma}) = (\mathcal{F}(\sigma_1), \ldots, \mathcal{F}(\sigma_n))$*;*
- *(b) for each strategy* σ *for player* i *in* G*, there exists a polynomial-time TM* M^σ *that, given as input* 1^n *and a view* v *for player* i *in* G_n *that is reachable when player* i *plays* $\mathcal{F}(\sigma_i)$ *in* G_n*, returns an action for player* i *such that* $LA(f_n(\mathcal{F}(\sigma)(1^n, v, r_T))) = M^\sigma(1^n, v, r_T)$*, where* r_T *is the random tape used (remember that the view contains the randomness used so far).*
- *(c) for all strategy profiles* $\vec{\sigma}$ *in* G*, all players* i*, and all polynomial-time strategies* M_i' *for player* i *in* $\mathcal{G}$*, there exists a sequence* $\sigma_1', \sigma_2', \ldots$ *of strategies for player* i *in* G *such that* $\{\phi^{G_n}_{(M_i', \mathcal{F}(\vec{\sigma}_{-i}))}\}_n$ *is computationally indistinguishable from* $\{\rho^G_{(\sigma_n', \vec{\sigma}_{-i})}\}_n$*.*

Definition 3.4 requires the existence of a sequence $\vec{f} = (f_1, f_2 \ldots)$ in UG2 and a function $\mathcal{F}$ in UG4. When we want to refer specifically to f and $\mathcal{F}$, $\mathcal{G}$ $\langle \vec{f}, \mathcal{F} \rangle$-represents G.

Note that UG2 requires that if h and h' are in the same information set in G_n, then $f_n(h)$ and $f_n(h')$ must be in the same information set in G. This means that we can view f_n as a map from information sets to information sets. However, it does *not* require the converse. As discussed above, in $\mathcal{G}$, an exponential-time player may be able to make distinctions between histories that cannot be made of the corresponding histories in the underlying game. We would like to be able to say that a polynomial-time player cannot distinguish h and h' if $f_n(h)$ and $f_n(h')$ are in the same information set. As we show later, these conditions allow us to make such a claim.

Also note that since the game is finite, to show UG4(a) and UG4(b) hold, it is enough to prove they hold for deterministic strategies. Given a mapping $\mathcal{F}$ that satisfies UG4(a) and (b) for deterministic strategies, we can extend it to mixed strategies in the obvious way: since a mixed strategy is just a probability distribution over finitely many deterministic strategies, it can be implemented by a TM that plays that probability distribution up to negligible precision over the corresponding mapping of the deterministic strategies (such an approximating distribution can be easily constructed in polynomial time). It is obvious that UG4(a) still holds. UG4(b) holds since using v and r_T, we can reconstruct which deterministic strategy σ' in the support σ was actually used to reach v, and then use the corresponding TM $M^{\sigma'}$.

3.2 The commitment game as a uniform computable sequence

We now consider how these definitions play out in the game G in Figure 1. Let $\mathcal{G} = (G_1, G_2, \ldots)$ be the sequence where G_n is the game where at the empty history player 1 uses a commitment scheme with an $(n-1)$-bit key, and outputs the commitment string as his action. Player 2 then plays either 0 or 1. Finally, player 1 outputs a string that is intended to be the commitment key. If he reveals the right key, and he committed to a bit different than what player 2 played, he wins. Otherwise, he loses.

Lemma 3.5. *$\mathcal{G}$ represents G.*

Proof. First, we show that $\mathcal{G}$ is a computable uniform sequence. All the games in the sequence involve exactly 2 players; the set of histories in G_n is a subset of $\{0,1\}^n$, and it is easy to compute the next player to act; finally, the utility functions are polynomial-time computable by using the TM R of the commitment scheme.

Next we show that the sequence represents G. There is an obvious mapping from histories of the games in the sequence to histories of G: a commitment to 0 is mapped to 0, a commitment to 1 is mapped to 1, the action of player 2 is just mapped to the action in G, player 1 providing the right key is mapped to action "open", and player 1 providing a wrong key is mapped to "destroy". Finally, it is easy to verify that UG3 (the condition on utilities) holds.

To show that UG4 holds, we need to define a function $\mathcal{F}$. A strategy for player 2 in G can't depend on player 1's action, since player 2's information set contains both actions. Thus, a deterministic strategy σ_2 for player 2 in G just plays an action in $\{0,1\}$; the corresponding strategy $\mathcal{F}(\sigma_2)$ just plays the same string. UG4(b) holds trivially in this case. To define $\mathcal{F}(\sigma_1)$ for a strategy σ_1 for player 1, we need to show how to implement each action of player 1. To play b at the empty history in G_n, 1 plays the commitment string $C_1(1^n, b, r)$, where r is the random string used. To play the action "open", it computes $k = C_2(1^n, b, r)$; to play "destroy", it plays $k \oplus 1$ (a string other than the right key). It is easy to see that UG4(b) holds for strategies of player 1. Moreover, it is easy to see that $\mathcal{F}(\vec{\sigma})$ corresponds to $\vec{\sigma}$, so UG4(a) holds. We extend $\mathcal{F}$ to mixed strategies as described above.

To see that UG4(c) holds, observe that a strategy for player 1 in G_n can clearly be mapped to a strategy in G: At the empty history player 1 has some distribution over commitments to 0 and commitments to 1. This clearly maps to a distribution over putting 0 and 1 in the envelope. At the other nodes where player 1 moves, G_n induces a distribution over correctly revealing the

commitment or doing some other action; again, this clearly maps to a distribution over "open" and "destroy" in the obvious way. Since a strategy M_1' for player 1 in $\mathcal{G}$ induces, for all n, a strategy $M_{1,n}'$ for player 1 in G_n, we can associate a sequence $(\sigma_1', \sigma_2', \ldots)$ with M_1'. It is easy to check that, for all strategies σ_2 for player 2 in G, $\{\phi^{G_n}_{(M_i', \mathcal{F}(\sigma_2))}\}_n$ is computationally indistinguishable from $\{\rho^G_{(\sigma_n', \sigma_2)}\}_n$.

We similarly want to associate with each strategy for player 2 in $\mathcal{G}$ a sequence of strategies in G. This is a little more delicate, since the information structure in G_n is not the same as that in G. Given a strategy σ_1 for player 1 in G, and an arbitrary polynomial-time strategy M_2 for player 2 in $\mathcal{G}$, let $P_n(b)$ the probability that M_2 plays b when $(\mathcal{F}(\sigma_1), M_2)$ is played in G_n. Let σ_n' be the strategy in G that plays according to the same distribution. We now claim that $\{\phi^{G_n}_{(\mathcal{F}(\sigma_1), M_2)}\}_n$ is indistinguishable from $\{\rho^G_{\sigma_1, \sigma_n'}\}_n$. Assume, by way of contradiction, that it is not. This can happen only if, for infinitely many n, M_2 plays 0 and 1 with non-negligibly different probabilities, depending on whether it is faced with a commitment to 0 or a commitment to 1. But that means that, for infinitely many n, it can distinguish those two events with non-negligible probability. This contradicts the assumption that the commitment scheme is secure. □

3.3 Computational information sets

In this section, we discuss the connection between computational indistinguishability and information structure in games. As we saw, when going from the game G in Figure 1 to the game $\mathcal{G}$ that represents it, we replaced the information set in G (the use of an envelope) with computational indistinguishability (a commitment scheme). Although the games in $\mathcal{G}$ are perfect information games, so that the players have complete information about a history, if player 1 uses the commitment scheme appropriately, then player 2 does not really understand the "meaning" of a history (i.e., whether it represents a commitment to 0 or a commitment to 1). On the other hand, if player 1 "cheats" by using, for example, some low-entropy random string for the commitment, player 2 might have a strategy that is able to understand the "meaning" of its action. Thus, there is a sense in which the information structure of a computational game depends on the strategies of the players. This dependence on strategies does not exist in standard games. If each of two histories h and h' in some information set I for player i has a positive probability of being reached by a particular strategy profile, then when player i is in I, he will not know which of h or h' was played, even if he knows exactly what strategies are being played. The situation is different for computational games, in a way we now make precise.

Suppose that $\mathcal{G} = (G_1, G_2, \ldots)$ $\langle \vec{f}, \mathcal{F} \rangle$-represents G and h is a history of G, so that $f_n^{-1}(h)$ is the set of histories of G_n that are mapped to h by f_n. For a set H of histories of a game $G_n \in \mathcal{G}$, let $\mathcal{V}_n(H)$ be the set of views that a player can have at histories in H when G_n is played. For a strategy profile $\vec{M}$ in $\mathcal{G}$, let $\xi^{G_n}_{\vec{M}}(v)$ be the probability of reaching view $v \in \mathcal{V}_n(H)$ if the players play strategy profile $\vec{M}$ in G_n. For a set V of views, let $\xi^{G_n}_{\vec{M}}(V) = \sum_{v \in V} \xi^{G_n}_{\vec{M}}(v)$. For a set V of mutually incompatible views (i.e., a set V of views such that for all distinct views $v, v' \in V$, the probability of reaching v given that v' is reached is 0, and vice versa), let $X^V_{\vec{M},n}$ be a probability distribution on V such that $X^V_{\vec{M},n}(v) = \frac{\xi^{G_n}_{\vec{M}}(v)}{\xi^{G_n}_{\vec{M}}(V)}$ if $\xi^{G_n}_{\vec{M}}(V) > 0$, and $\frac{1}{|V|}$ otherwise. Let $\xi^G_{\vec{\sigma}}(S)$ denote the probability of reaching a set S of histories in G if the players play strategy profile $\vec{\sigma}$. Note that if $\xi^G_{\vec{\sigma}}(S) > 0$, then by UG4, for all sufficiently large n, we must have $\xi^{G_n}_{\vec{M}_{\vec{\sigma}}}(\mathcal{V}_n(f_n^{-1}(S))) > 0$.

We now define the notion of a computational information partition.

Definition 3.6. *Let $\mathcal{G}$ $\langle \vec{f}, \mathcal{F} \rangle$-represent a game G and let $\vec{M}$ be a strategy in $\mathcal{G}$. A partition $\mathcal{I}_i$ of H^i (recall that this is the set of histories in G where i plays) is* $\vec{M}$-consistent *for player i if, for all non-singleton $I \in \mathcal{I}_i$ and all $h \in I$ such that both $\xi^{G_n}_{\vec{M}}(\mathcal{V}_n(f_n^{-1}(h)))$ and $\xi^{G_n}_{\vec{M}}(\mathcal{V}_n(f_n^{-1}(I \setminus h)))$ are*

non-negligible, $\{X_{\vec{M},n}^{\mathcal{V}_n(f_n^{-1}(h))}\}_{n\in\mathbb{N}}$ *is computationally indistinguishable from* $\{X_{\vec{M},n}^{\mathcal{V}_n(f_n^{-1}(I\setminus\{h\}))}\}_{n\in\mathbb{N}}$. *A partition structure* $\vec{I}$ *is* $\vec{M}$*-consistent if, for all agents* i, $\vec{\mathcal{I}}_i$ *is* $\vec{M}$*-consistent*

Intuitively, a partition $\mathcal{I}_i$ for player i is consistent with a strategy profile $\vec{M}$, if, when $\vec{M}$ is played in $\mathcal{G}$, for all $I \in \mathcal{I}_i$ and all histories $h, h' \in I$, the distribution over views that map to h is computationally indistringuishable from the distribution over views that map to h'. In our example, this means that player 2 can't distinguish between the distribution created by a commitment to 0 and the distribution created by a commitment to 1.

Note that we do not enforce any condition on histories in G that are mapped back to a set of histories that is reached with only negligible probability. This means there might be more than one $\vec{M}$-consistent information partition.

We next show that if $\mathcal{I}_i$ is the information partition of player i in G, and $\mathcal{G}$ $\langle\vec{f},\mathcal{F}\rangle$-represent G then for any strategy profile $\vec{\sigma}$ in G, $\mathcal{I}_i$ must be $\mathcal{F}(\vec{\sigma})$-consistent.

Theorem 3.7. *If* $\mathcal{G}$ $\langle\vec{f},\mathcal{F}\rangle$*-represents* G, $\mathcal{I}_i$ *is the information partition of player* i *in* G, *and* $\vec{\sigma}$ *is a strategy profile in* G *then* $\mathcal{I}_i$ *is* $\mathcal{F}(\vec{\sigma})$*-consistent.*

Proof. We must show that if $I \in \mathcal{I}_i$ is a non-singleton information set for i in G and $h \in I$, then for all strategy profiles $\vec{\sigma}$ in G such that $\xi_{\vec{\sigma}}^G(h) > 0$ and $\xi_{\vec{\sigma}}^G(I \setminus \{h\}) > 0$, $\{X_{\mathcal{F}(\vec{\sigma}),n}^{\mathcal{V}_n(f_n^{-1}(h))}\}_{n\in\mathbb{N}}$ is computationally indistinguishable from $\{X_{\mathcal{F}(\vec{\sigma}),n}^{\mathcal{V}_n(f_n^{-1}(I\setminus\{h\}))}\}_{n\in\mathbb{N}}$.

Assume, by way of contradiction, that $h \in I$, I is an information set for player i in G, and there exists a strategy profile $\vec{\sigma}$ in G that reaches both h and $I \setminus \{h\}$ with positive probability such that $\{X_{\mathcal{F}(\vec{\sigma}),n}^{\mathcal{V}_n(f_n^{-1}(h))}\}_n$ is distinguishable from $\{X_{\mathcal{F}(\vec{\sigma}),n}^{\mathcal{V}_n(f_n^{-1}(I\setminus\{h\}))}\}_n$. Thus, there exists a distinguisher D for these distributions. Let a and a' be distinct actions in $A(I)$. (Recall that we assumed that $|A(I)| \geq 2$.) Let M' be a strategy for player i in $\mathcal{G}$ such that when M' reaches a history that maps to I (by UG4(b) and the fact that the sets of actions available in each information set are disjoint, this can be checked in polynomial time), M' uses D to distinguish if its view is in $\mathcal{V}_n(f_n^{-1}(h))$ or $\mathcal{V}_n(f_n^{-1}(I \setminus \{h\}))$. M' then plays an action mapped to a if D returns 0 and an action mapped to a' otherwise. At a history other than one in $f_n^{-1}(I)$, M' plays like $\mathcal{F}(\sigma_i)$. It is easy to see that, because $\{X_{\mathcal{F}(\vec{\sigma}),n}^{f_n^{-1}(h)}\}_n$ and $\{X_{\mathcal{F}(\vec{\sigma}),n}^{f_n^{-1}(I\setminus\{h\})}\}_n$ are distinguishable with non-negligible probability, there is a non-negligible probability that the strategy M' is able to detect which case holds, and play accordingly. That means that when histories of $(M', \mathcal{F}(\sigma_{-i})$ are mapped to histories of G via f_n, there is a non-negligible gap between the probability of (h, a) and the probability of (h', a) for $h' \in I \setminus \{h\}$. Since $h \in I$, there can be no strategy σ' for player i such that (σ', σ_{-i}) has such a gap, and UG4(c) cannot hold. This gives us the desired contradiction. □

Note that Theorem 3.7 holds trivially if, for all $G_i \in \mathcal{G}$, all the histories of $\mathcal{G}$ that map to I are in the same information set in G_i. The theorem is of interest only when this is not the case. This result can be thought of saying that there are information sets in G that model real lack of information and information sets in G that model computational indistinguishability.

4 Solution Concepts for Computational Games

In this section, we consider analogues of two standard solution concepts in the context of computational games: Nash equilibrium and sequential equilibrium, and prove that they exist.

4.1 Computational Nash equilibrium

Informally, a strategy profile in $\mathcal{G}$ is a computational Nash equilibrium if no player i has a profitable *polynomial-time* deviation, where a deviation is taken to be profitable if it is profitable in infinitely many games in the sequence. Recall that $\psi_{\vec{M}}^{G_n}$ is the distribution on the terminal histories in G_n induced by a strategy profile $\vec{M}$ in $\mathcal{G}$.

Definition 4.1. *$\vec{M} = \{M_1, \ldots, M_c\}$ is a* computational Nash equilibrium *of a computable uniform sequence $\mathcal{G}$ if, for all players $i \in [c]$ and for all polynomial-time strategies M_i' in $\mathcal{G}$ for player i, there exists a negligible function ϵ, such that for all n,*

$$\sum_{h \in H_n^T} \psi_{\vec{M}}^{G_n}(h) u_i(h) \geq \sum_{h \in H_n^T} \psi_{(M', \vec{M}_{-i})}^{G_n}(h) u_i(h) - \epsilon(n).$$

Our definition of computational NE differs from the standard definition of ϵ-NE in two ways. First, we restrict to polynomial-time deviations. This seems in keeping with our focus on polynomial-time players. Second, we have a negligible loss of utility ϵ in the definition, and ϵ depends on the deviation. (The fact that ϵ depends on the deviation means that what we are considering cannot be considered an ϵ-Nash equilibrium in the standard sense.) Of course, we could have given a definition more in the spirit of the standard definition of Nash equilibrium by simply taking ϵ to be 0. However, the resulting solution concept would simply not be very interesting, given our restriction to polynomial-time players. In general, there will not be a "best" polynomial-time strategy; for every polynomial-time TM, there may be another TM that is better and runs only slightly longer. For example, player 2 may be able to do a little better by spending a little more time trying to decrypt the commitment in a commitment scheme. (See also the examples in [8].)[4]

We now show that if a computational game $\mathcal{G}$ represents G, then for every NE $\vec{\sigma}$ in G, there is a corresponding NE in $\mathcal{G}$.

Theorem 4.2. *If $\mathcal{G}$ $\langle \vec{f}, \mathcal{F} \rangle$-represents G and $\vec{\sigma}$ is a NE in G, then $\mathcal{F}(\vec{\sigma})$ is a computational NE of $\mathcal{G}$.*

Proof. Suppose that $\vec{\sigma}$ is a NE in G. By UG4, $\vec{\sigma}$ corresponds to $\mathcal{F}(\vec{\sigma})$. Thus, there exists some negligible function ϵ such that, for all n,

$$\sum_{h \in H^T} \phi_{\mathcal{F}(\vec{\sigma})}^{G_n}(h) u_i(h) > \sum_{h \in H^T} \rho_{\vec{\sigma}}^{G}(h) u_i(h) - \epsilon(n).$$

We claim that $\vec{M}_{\vec{\sigma}}$ is a computational NE of $\mathcal{G}$. Assume, by way of contradiction, that it is not. That means there is some player i, some strategy M_i' for player i, and some constant $c > 0$ such that, for infinitely many values of n,

$$\sum_{h \in H^T} \phi_{(M', \mathcal{F}(\vec{\sigma}_{-i}))}^{G_n}(h) u_i(h) > \sum_{h \in H^T} \phi_{\mathcal{F}(\vec{\sigma})}^{G_n}(h) u_i(h) + \frac{1}{n^c};$$

If not, we could have constructed a negligible function to satisfy the equilibrium condition.

By combining the two equation we get that for infinitely many values of n,

$$\sum_{h \in H^T} \phi_{(M', \mathcal{F}(\vec{\sigma}_{-i}))}^{G_n}(h) u_i(h) > \sum_{h \in H^T} \rho_{\vec{\sigma}}^{G}(h) u_i(h) - \epsilon(n) + \frac{1}{n^c}.$$

[4] One way to avoid having ϵ depend on the deviation, which we do not explore in this paper, is to instead use a concrete model of complexity in which we use only TMs with running time less than some specific function T of n. In that case, we could use a single global ϵ, and get a definition which is closer to that of traditional ϵ-NE.

Since $\vec{\sigma}$ is a NE, we get that for all sequences $\sigma_1', \sigma_2' \ldots$ of strategies for player i in G,

$$\sum_{h \in H^T} \rho^G_{\vec{\sigma}}(h) u_i(h) \geq \sum_{h \in H^T} \rho^G_{(\sigma_n', \vec{\sigma}_{-i})}(h) u_i(h).$$

This means that for infinitely many values of n, and for any such sequence,

$$\sum_{h \in H^T} \phi^{G_n}_{(M', \mathcal{F}(\sigma_{-i}))}(h) u_i(h) > \sum_{h \in H^T} \rho^G_{(\sigma_n', \vec{\sigma}_{-i})}(h) u_i(h) - \epsilon(n) + \frac{1}{n^c}.$$

But this contradicts UG4(c), which says that there must exist a sequence $\sigma_1', \sigma_2' \ldots$ such that $\{\phi^{G_n}_{(M_i', \mathcal{F}(\vec{\sigma}_{-i}))}\}_n$ is computationally indistinguishable from $\{\rho^G_{(\sigma_n', \vec{\sigma}_{-i})}\}_n$. Since the difference between the two payoffs is not negligible, a distinguisher could just sample enough outcomes of these strategies and compute the average payoff to distinguish the two distributions with non-negligible probability. Thus, $\vec{M}_{\vec{\sigma}}$ must be a computational NE of $\mathcal{G}$. □

Theorem 4.2 shows to every NE in G there is a corresponding NE in $\mathcal{G}$. The converse does not hold. This should not be surprising; the set of strategies in $\mathcal{G}$ is much richer than that in G. The following example gives a simple illustration.

Example. Consider the 2-player game G' that is like the game in Figure 1, except that the payoff is 1 to both if they match and 0 otherwise (and both get -1 if player 1 does not open the envelope). This game has three NE: both play 0; both play 1; and both play the mixed strategy that gives probability 1/2 to each of 0 and 1. There is a computational game $\mathcal{G}'$ that represents G' that is essentially identical to the game $\mathcal{G}$ described in Section 3.2, except that the payoffs are modified appropriately. The game $\mathcal{G}'$ has many more equilibria than G', since player 1 can commit to 0 and 1 with 0.5 probability but use a fixed key that the second player knows (or choose a random key from a low entropy set that the second player can enumerate). Player 2 can take advantage of this to always play the matching action. There is no strategy in G' that can mimic this behavior.

4.2 Computational sequential equilibrium

Our goal is to define a notion of computational sequential equilibrium. To do so, it is useful to think about the standard definition of sequential equilibrium at an abstract level. Essentially, $\vec{\sigma}$ is a sequential equilibrium if, for each player i, there is a partition $\mathcal{I}_i'$ of the histories where i plays such that, at each cell $I \in \mathcal{I}_i'$, player i has beliefs about the likelihood of being at each history in I, and the action that he chooses at a history in I according to σ_i is a best response, given these beliefs and what the other agents are doing (i.e., σ_{-i}). The standard definition of sequential equilibrium takes the partition $\mathcal{I}_i'$ to consist of i's information sets. If we partition the histories into singletons, we get a *subgame-perfect equilibrium* [**?**]. As we argued in Section **??**, the information sets sets in $\mathcal{G}$ are too fine, in general, to capture a player's ability to distinguish. Thus, as a first step to getting a notion of computational sequential equilibrium, we generalize the standard definition of sequential equilibrium in a straightforward way to get $\vec{\mathcal{I}}$-sequential equilibrium, where $\mathcal{I}_i$ is an arbitrary partition of the histories where i plays.

Definition 4.3. *Given an partition $\vec{\mathcal{I}}$, $\vec{\sigma}$ is a* $\vec{\mathcal{I}}$-sequential equilibrium *of G if there exists a sequence of completely mixed strategy profiles $\vec{\sigma}^1, \vec{\sigma}^2, \ldots$ converging to $\vec{\sigma}$ and a sequence $\delta_1, \delta_2, \ldots$ of nonnegative real numbers converging to 0 such that, for each player i and each set $I \in \mathcal{I}_i$, $\vec{\sigma}_i^n$ is a δ_n-best response to $\vec{\sigma}_{-i}^n$ conditional on having reached I.*

What are reasonable partition structures to use when considering a computational game? As we suggested, using the information partition structure of $\mathcal{G}$ seems unreasonable. For example,

in our example commitment game, this does not allow the second player to act the same when facing commitment to 0 and commitments to 1, although, as we argued earlier, if player 1 plays appropriately, computationally bounded player cannot distinguish these two events.

It seems reasonable to have histories in the same cell of the partition if the player cannot distinguish what these histories actually "represent". For general uniform computable sequences it is unclear what "represents" should mean. However, if $\mathcal{G}$ represents a game G, then we do have in some sense a representation for a history: the history it maps to in the underlying game. As we saw in Section 3.3, what a player can infer from a history might depend not just on the information partition structure of the games in $\mathcal{G}$, but also on the strategies played by the players in G. Thus, a natural candidate for a partition structure $\vec{I}$ when $\vec{M}$ are the strategies played, is a partition that is based on a $\vec{M}$-consistent partition structure $\vec{\mathcal{I}}_G$ of the histories of G. We now formalize this intuition.

Suppose that $\mathcal{G}$ $\langle \vec{f}, \mathcal{F} \rangle$-represents G. Given a set $I \subseteq H$, let I_{G_n} be the set consisting of histories $h \in G_n$ such that $f_n(h) \in I$. Given two strategies M and M' for a player in $\mathcal{G}$, let (M, I, M') be the TM that plays like M in G_n up to I_{G_n}, and then switches to playing M' from that point on. For a game $G_n \in \mathcal{G}$, a strategy profile $\vec{M}$, and a set H'_n of histories in G_n that is reached with positive probability when $\vec{M}$ is played, let $\phi^{G_n}_{\vec{M},H'_n}$ be the probability on terminal histories in G induced by pushing forward the probability on terminal histories in G_n conditioned on reaching H'_n (where we identify the event "reaching H'_n" with the set of terminal histories that extend a history in H'_n). We can similarly define $\rho^G_{\vec{\sigma},H'}$ for a subset H' of histories in G.

Definition 4.4. *Suppose that $\mathcal{G}$ $\langle \vec{f}, \mathcal{F} \rangle$-represents G. Then $\vec{M} = \{M_1, \ldots, M_c\}$ is a* computational sequential equilibrium *of $\mathcal{G}$ if there exists a sequence of completely mixed strategies $\vec{M}^1, \vec{M}^2, \ldots$ converging to $\vec{M}$ and a sequence $\delta_1, \delta_2, \ldots$ converging to 0 such that, for all k, n, and players $i \in [c]$, there exists a $\vec{M}$-consistent partition $\mathcal{I}_i$ such that, for all sets $I \in \mathcal{I}_i$ and all polynomial-time strategies M' for player i in $\mathcal{G}$, there exists a negligible function ϵ such that*

$$\sum_{h \in H^T} \phi^{G_n}_{\vec{M}^k, I_{G_n}}(h) u_i(h) \geq \sum_{h \in H^T} \phi^{G_n}_{((\vec{M}^k_i, I, M'), \vec{M}^k_{-i}), I_{G_n}}(h) u_i(h) - \epsilon(n) - \delta_k.$$

We now prove that, as with NE, if $\vec{\sigma}$ is a sequential equilibrium of an extensive form game G with perfect recall and $\mathcal{G}$ represents G, then $\mathcal{F}(\vec{\sigma})$ is a computational sequential equilibrium of $\mathcal{G}$.

Theorem 4.5. *Suppose that $\mathcal{G}$ $\langle \vec{f}, \mathcal{F} \rangle$-represents G and G has perfect recall. If there exists a belief function μ such that $(\vec{\sigma}, \mu)$ is a sequential equilibrium in G, then $\mathcal{F}(\vec{\sigma})$ is a computational sequential equilibrium of $\mathcal{G}$.*

Proof. Suppose that there exists a belief system μ such that $(\vec{\sigma}, \mu)$ is a sequential equilibrium in G. Thus, there exists a sequence of completely mixed strategy profiles $\vec{\sigma}^1, \vec{\sigma}^2, \ldots$ that converges to $\vec{\sigma}$ and a sequence $\delta_1, \delta_2, \ldots$ that converges to 0 such that for all players i, all information sets I for i in G, and all strategies σ' for i that act like σ on all prefixes of histories in I, we have that

$$\sum_{h \in H^T} \rho^G_{\vec{\sigma}^k, I}(h) u_i(h) \geq \sum_{h \in H^T} \rho^G_{(\sigma', \vec{\sigma}^k_{-i}), I}(h) u_i(h) - \delta_k. \tag{1}$$

Assume, by way of contradiction, that $\vec{M} = \mathcal{F}(\vec{\sigma})$ is not a computational sequential equilibrium. Let M^k_i be the TM that acts like $\mathcal{F}(\sigma^k_i)$ except that at a view it is called to play, with some negiligible probability, it plays an arbitrary legal action, chosen uniformly at random. Note that this makes M^k_i completely mixed, while ensuring that $\vec{M}^k$ still corresponds to $\vec{\sigma}^k$. Also note that the sequence $\vec{M}^1, \vec{M}^2, \ldots$ converges to $\vec{M}$. By Theorem 3.7, if $\mathcal{I}_i$ is the information partition of player i in G,

then $\mathcal{I}_i$ is $\vec{M}^k$-consistent for all k and, in particular, is also $\vec{M}$-consistent. That means that there is some k, player i, information set I for i in G, strategy M_i' for i, and constant c such that, for infinitely many values of n,

$$\sum_{h \in H^T} \phi^{G_n}_{((\vec{M}_i^k, I, M'), \vec{M}^k_{-i}), I_{G_n}}(h) u_i(h) > \sum_{h \in H^T} \phi^{G_n}_{\vec{M}^k, I_{G_n}}(h) u_i(h) + \frac{1}{n^c} + \delta_k. \tag{2}$$

If not, $\vec{M}$ would be a computational sequential equilibrium.

Since $\vec{\sigma}^k$ is completely mixed, every terminal history is reached with positive probability. Thus, I_{G_n} is reached with positive probability. Since $\vec{M}^k$ corresponds to $\vec{\sigma}^k$, $\{\phi^{G_n}_{\vec{M}^k, I_{G_n}}\}_n$ (the conditional ensemble) must be statistically close to $\{\rho^G_{\vec{\sigma}^k, I}\}_n$, for otherwise we could use the distinguisher for these ensembles to distinguish the unconditional ensembles. It follows that there exists some negligible function ϵ such that, for all n,

$$\sum_{h \in H^T} \phi^{G_n}_{\vec{M}^k, I_{G_n}}(h) u_i(h) > \sum_{h \in H^T} \rho^G_{\vec{\sigma}^k, I}(h) u_i(h) - \epsilon(n). \tag{3}$$

From (2) and (3), it follows that, for infinitely many values of n,

$$\sum_{h \in H^T} \phi^{G_n}_{((\vec{M}_i^k, I, M'), \vec{M}^k_{-i}), I_{G_n}}(h) u_i(h) > \sum_{h \in H^T} \rho^G_{\vec{\sigma}^k, I}(h) u_i(h) - \epsilon(n) + \frac{1}{n^c} + \delta_k. \tag{4}$$

By UG4(c), there is a sequence $\sigma_1', \sigma_2', \ldots$ of strategies for i in G such that $\{\phi^{G_n}_{((\vec{M}_i^k, I, M'), \vec{M}^k_{-i}})\}_n$ is computationally indistinguishable from $\{\rho^G_{(\sigma_n', \vec{\sigma}^k_{-i})}\}_n$. Since, for n sufficiently large, I_{G_n} is reached with non-negligible probability by $\vec{M}^k$, and $(\vec{M}_i^k, I, M')$ acts like $\vec{M}_i^k$ in all prefixes of histories in I_{G_n}, it must be the case that for n sufficiently large, $((\vec{M}_i^k, I, M'), \vec{M}^k_{-i})$ reaches I_{G_n} with non-negligible probability. Moreover, $\{\phi^{G_n}_{((\vec{M}_i^k, I, M'), \vec{M}^k_{-i}), I_{G_n}}\}_n$ is computationally indistinguishable from $\{\rho^G_{(\sigma_n', \vec{\sigma}^k_{-i}), I}\}_n$. If not, again, a distinguisher for the unconditional distributions can just use the distinguisher for the conditional distribution by calling it only when the sampled history is such that I is visited. From (1) and (4), we get that for infinitely many values of n,

$$\sum_{h \in H^T} \phi^{G_n}_{((\vec{M}_i^k, \mathcal{I}(I), M'), \vec{M}^k_{-i}), I_{G_n}}(h) u_i(h) > \sum_{h \in H^T} \rho^G_{(\sigma_n', \vec{\sigma}^k_{-i}), I}(h) u_i(h)(h) u_i(h) - \epsilon(n) + \frac{1}{n^c}.$$

But, as in previous arguments, this contradicts the assumption that $\{\phi^{G_n}_{((\vec{M}_i^k, I, M_i'), \vec{M}^k_{-i}), I_{G_n}}\}_n$ is computationally indistinguishable from $\{\rho^G_{(\sigma_n', \vec{\sigma}^k_{-i}), I}\}_n$. Thus, $\vec{M}_{\vec{\sigma}}$ is a computational sequential equilibrium of $\mathcal{G}$. □

What are the beliefs represented by this equilibrium? The beliefs we get are such that the players believe that only strategies that are mappings (via $\mathcal{F}$) of strategies in the underlying game have been used, so they explain deviations in the computational game in terms of deviations in the underlying game.

By considering a consistent partitions here, we effectively average the expected payoff over all histories of $\mathcal{G}_n$ that map to the same information set in I. Note that, for each specific history in this set, there might be a better TM. For example, in the commitment game discussed before, for each commitment string, there is a TM for player 2 that does better then the prescribed protocol: the one that plays the right value given that string. However, our notion considers the expected

value over all these histories, and thus a good deviation does not exist. Since no polynomial-time TM can tell to which histories in the underlying game these histories are mapped (via f), we treat cells in a consistent partition just as traditional information sets are treated in the standard notion of sequential equilibrium.

Theorems 4.2 and 4.5 show that we may be able to carry out an analysis of rationality in cryptographic protocols by analyzing rationality in the much simpler abstract game underlying them. For example, it is now easy to see that the strategy profile in which player 1 commits to 0 with probability 0.5 and always opens and player 2 also plays 0 with probability 0.5 is a sequential equilibrium in the computational game from Section 3.2.

5 Stateful TMs vs. Stateless TMs

As we observed in the introduction, the question of whether stateful or stateless TMs are used played a significant role in distinguishing the positive results of [10] about the existence of a polynomial-time computable NE in repeated games from the negative results of [2] showing that, in general, we cannot compute a NE in repeated games in polynomial time. In this paper, we use stateful TMs, since they seem to be needed to implement mixed strategies. But are they really needed? As the following result shows, they are (at least, given standard cryptographic assumptions).

Consider again underlying game from Section 3.2, but now with a different computational game that represents it: now take the computational game $\mathcal{G}'$ to be such that the length of an action in G_n is at most $n + \log(n)$ (rather than n), and the first $\log(n)$ bits of the first action determine the length of the key used by the commitment scheme. For a strategy σ for player 1, $\mathcal{F}(\sigma)$ plays a commitment scheme whose length is that exactly encoded of the number represented (in binary) by the first $\log(n)$ bits of its first output. With this small change, an argument like that used in Section 3.2 shows that $\mathcal{G}'$ represents G. The game G has the same obvious NE as before: both players play 0 and 1 with equal probability, and player 1 then plays "open". the corresponding strategies form a computational NE in $\mathcal{G}$. By Theorem 4.2, the corresponding strategies form a computational NE in $\mathcal{G}'$.

This computational NE uses stateful TMs. The following result shows that we cannot implement any computational NE of $\mathcal{G}$ using stateless TMs. To get this result, we need to assume the existence of an *exponentially hard* commitment scheme. This means that there exists a constant $c > 0$ such that, for all k, no algorithm with running time 2^{ck} can distinguish a commitment to 0 from a commitment to 1 with probability greater than $\frac{1}{2^{ck}}$.

Theorem 5.1. *There is no computational NE of $\mathcal{G}$ consisting of stateless TMs.*

Proof. Assume, by way of contradiction, that (M_1, M_2) is a computational NE of $\mathcal{G}$ where M_1 and M_2 are stateless. We first claim that player 2 can obtain a guaranteed payoff of at least $1 - \frac{2}{n}$. Consider the following TM M_2'. Given as input a history h, M_2' simulates M_1 on input $(h, 1)$ n^2 times. Since M_1 gets only a view as input, its actions can be simulated by M_2'; since M_2 can compute M_1's view. (Note that this would not be the case if M_1 were stateful, for then its action at the second step could depend on the randomness it used at the first step, and M_2' would not have access to this.) Since M_1 is a polynomial-time TM, so is M_2'. If in any of these simulations M_1 outputs the valid commitment key (M_2' can easily check if the key is valid by using the TM R), then M_2' knows what bit was committed to by M_1, and plays that as its next action, guaranteeing that player 2 wins. Otherwise, M_2 plays 1.

Assume first that M_1 is a TM that on input $(h, 1)$ reveals the valid key with probability less than $1/n$. This means that by playing 1, M_2' wins with probability at least $1 - \frac{1}{n}$, and thus its expected payoff is at least $1 - \frac{2}{n}$. Otherwise, the probability that M_1 reveals the valid key is greater than $\frac{1}{n}$, and thus in n^2 simulation of M_1, the probability that the valid key is not revealed is less

than $(1-\frac{1}{n})^{n^2}$. Thus, player 2's payoff is at least $1-(1-\frac{1}{n})^{n^2} > 1-\frac{2}{n}$, so player 2 has a deviation that gives him an expected payoff of at least $1-\frac{2}{n}$.

We next show that player 1 can obtain a guaranteed payoff of at least $\frac{1}{2}-\frac{1}{n}$. Let $a > 1$ be some constant such that M_2's running time is bounded by n^a. Let c be the constant from the exponentially hard commitment scheme definition. Let n_0 be such that $(\frac{a}{c})\log n < n-1$ for $n > n_0$. Consider the following TM M_1'. If $n < n_0$, in G_n, M_1' behaves just like M_1. If $n \geq n_0$, at the empty history, M_1' uses a commitment scheme with key length $\frac{a}{c}\log n$ to commit to a bit chosen uniformly at random. At the next history that player 1 is called upon to act, M_1' checks each string of length $\frac{a}{c}\log n$ to see if it is a valid key for the commitment; if so, it outputs that string. Since c and a are constants, this can be done in polynomial time, and thus M_1' is a valid deviation.

We now analyze the expected utility of M_1' in G_n for $n \geq n_0$. Since M_1' enumerates all possible strings that it might have used in the first step, we know that one of them must work, and thus M_1' succeeds in opening the commitment. Thus, M_1' loses only if M_2 plays the same bit as M_1' committed to. We claim that this can't happen with probability greater than $\frac{1}{2}+\frac{1}{n}$. Assume otherwise. Then the probability that M_2 plays 1 when M_1 commits to 0 must differ from the probability that M_2 plays 1 when M_1 commits to 1 by at least $\frac{2}{n}$. But this means that M_2 can be used as a distinguisher that runs in time n^a and distinguishes with probability $\frac{2}{n}$, which contradictions the assumption that the commitment scheme is exponentially hard.

This means that with M_1', player 1 can get at least $1-(\frac{1}{2}+\frac{1}{n})$. Thus, if (M_1, M_2) is a computational NE for $\mathcal{G}$, then there must exist a negligible function ϵ such that the expected payoff of player 1 in G_n is at least $1-\frac{2}{n}-\epsilon(n)$ and the expected payoff of player 2 in G_n is at least $\frac{1}{2}-\frac{1}{n}-\epsilon(n)$. Since the combined expected payoff of the two players is at most 1, we have that $\frac{3}{2}-\frac{3}{n}-2\epsilon(n) < 1$. But this means that $\epsilon(n) \geq \frac{1}{4}-\frac{3}{2n}$, so ϵ is non-negligible. □

We note that when the players are not computationally bounded, a stateless TM can always implement a stateful TM. For example, whenever it is called on to play, it can enumerate all possible random strings that the stateful TM has used thus far, and select uniformly at random among the ones consistent with the history of play so far.

6 Application: Implementing a Correlated Equilibrium Without a Mediator

In this section, we show that our approach can help us analyze protocols that use cryptography to implement a correlated equilibrium (CE) in a normal-form game. Dodis, Halevi, and Rabin [3] (DHR) were the first to use cryptographic techniques to implement a CE. They did so using a protocol that they showed was a NE, provided that players are computationally bounded (for a notion of computational NE that is related to ours). However, as discussed by Gradwohl, Livne, and Rosen [6] (GLR), DHR's proposed protocol does not satisfy solution concepts that also require some sort of sequential rationality. DHR's protocol punishes deviations using a minimax strategy that may give the punisher as well the player being punished a worse payoff; thus, it is just an empty threat. To deal with this issue, GLR introduce a solution concept that they call *Threat Free Equilibrium (TFE)*, which explicitly eliminates such empty threats. GLR additionally provide a protocol that can implement a CE in a normal-form game that is a convex combination of NEs (CCNE), without using a mediator; the GLR protocol is a TFE if the players are computationally bounded.

We now show provide a protocole similar in spirit to the one used in GLR that implements a CE that is a CCNE; our protocol is a computational sequential equilibrium if the players are computationally bounded. Unlike GLR, we are able to apply our appraoch to CEs in games with more than 2 players. We require that the CCNE is of finite support, that all its coefficients are

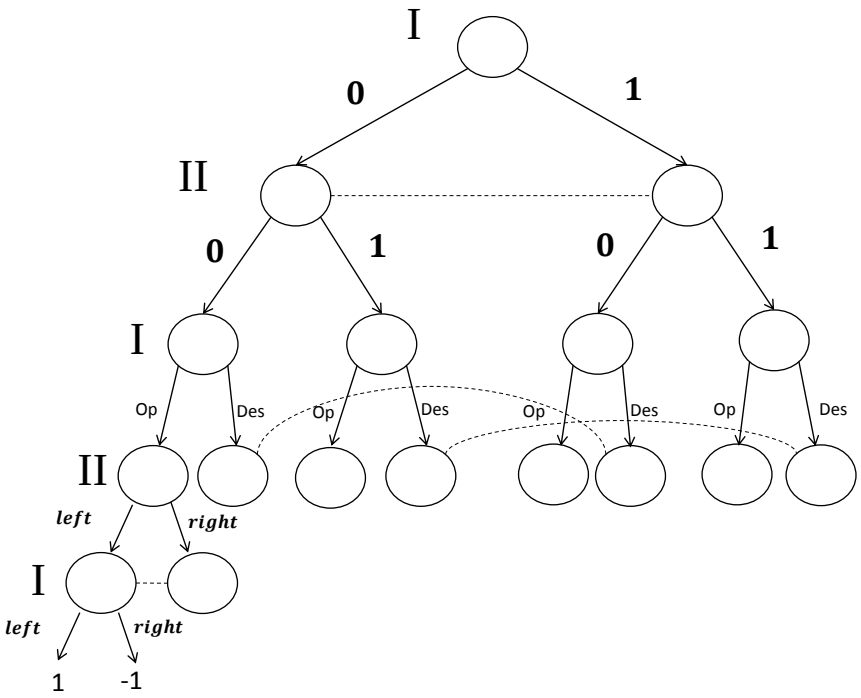

Figure 2: An example of the game G_{corr} where $\ell = 2$ and G is a coordination game

rational numbers, and that each of the NEs in its support has coefficients that are rational numbers.[5] We call such CCNEs *nice*. Note that any CCNE can be approximated to arbitrary accuracy by a nice CCNE.

Given a game G with a nice CCNE π, we show how to convert it to an extensive-form game G_{corr} that implements this CE without using crptography, but using envelopes; that is, G_{corr} has a sequential equilibrium with the same distribution over outcomes in G as π. We then show how to implement G_{corr} as a computational game using a cryptographic protocol.

Given G and π, let ℓ be the least common denominator of the coefficients of π. Let G_{corr} be the game where player 1 first puts an element of $\{0, \ldots, \ell - 1\}$ in an envelope, then player 2 plays an element in $\{0, \ldots, \ell - 1\}$ without knowing what player 1 played (all the histories where player 2 makes his first move are in the same information set of player 2). Then player 1 can either open the envelope or destroy it. All the histories after player 1 opens the envelope form singleton information for player 2; all histories after player 1 destroys the envelope and 2 initially played j are in the same information set for player 2, for $j \in \{0, \ldots, \ell - 1\}$. Then G is played. (Note that G might involve many players other than 1 and 2, but 1 and 2 are the only players who play in the initial part of G_{corr}.) The players move sequentially: first player 1 moves, then player 2 moves (without knowing player 1's move), then player 3 moves (with knowing 1 and 2's moves), and so on. The payoffs of G_{corr} depend only on the players' moves when playing the G component of G_{corr}, and are the same as the payoffs in G. See Figure 2 for a game G_{corr} when ℓ is 2 and G is a coordination game: that is, in G, each player moves either left or right, and each gets a payoff of 1 if they make the same move, and -1 if they make different moves.

Let σ be a NE in G in which player 1's payoff is no better than it is in any other NE in G.

[5] GLR required a stronger condition; they required all the coefficients to be rational numbers that are a power of two.

Now consider the following simple strategies for the players in G_{corr}. Intuitively, the players start by picking an NE in the support of π to play, with probability proportional to its coefficient in π. To this end, fix an ordering of length ℓ of the NEs in the support of π, where each NE appears a number of times proportional to its weight in the convex combination that makes up π. At the empty history, player 1 selects an action a uniformly at random from $\{0, \ldots, \ell - 1\}$ and puts it in the envelope. Then player 2 also selects an action b uniformly at random from $\{0, \ldots, \ell - 1\}$. Then player 1 opens the envelope. The players then play the NE in place $(a + b \mod \ell)$ in of NEs. If player 1 does not open, the players play according to σ. Call the resulting strategy profile $\vec{\sigma}_\pi$ It is not hard to verify that $\vec{\sigma}_\pi$ implements the CCNE, and that there exists beliefs μ such that $(\vec{\sigma}_\pi, \mu)$ is a sequential equilibrium of G_{corr}.

So now all we have to provide a computational game $\mathcal{G}_{corr}$ that represents G_{corr}, where the games in $\mathcal{G}_{corr}$ use cryptography instead an envelope for the first part of the game. Let d be such that $2^{d-1} \leq \ell < 2^d$. Let $\mathcal{G}_{corr}$ be the sequence where G_n is the game where, at the empty history, player 1 commits to a d-bit string by using d commitments in parallel, each with key length $n - 1$ and outputs the d commitment strings as his action. Player 2 then plays a bitstring of length d that can be viewed as a binary representation of a number in $\{0, \ldots, \ell - 1\}$. Player 1 then plays a string that is intended to be the commitment keys of the d commitments. Then the players play a string representing their action in G (again using its binary representation).

Theorem 6.1. *$\mathcal{G}_{corr}$ represents G_{corr}.*

Proof. It is obvious that $\mathcal{G}_{corr}$ is a computable uniform sequence. We now show that it represents G_{corr}. The mappings $\vec{f}$ of histories maps player 1's commitments to a string s to the action $s \mod \ell$. (Notice that the fact we used d commitments in parallel does not change the fact that the commitments are perfectly binding and thus this is well defined.) Actions of player 2 are mapped to an action $s \mod \ell$ according to their binary representation; if player 1 reveals d valid keys in h, then in $f_n(h)$ he plays "open", and otherwise he plays "destroy"; the actions of G are mapped in the obvious way.

To show that UG4 holds, we proceed as follows: The mapping $\mathcal{F}$ for any player other than 1 and 2 is obvious: he does the same thing in G, G_{corr}, and $\mathcal{G}_{corr}$. For player 2, note that player 2's first action in G_{corr} can't depend on player 1's action, since player 2's information set contains all the histories. Thus, a deterministic strategy σ_2 for player 2 in G_{corr} just plays an action in $\{0, \ldots, \ell-1\}$; $\mathcal{F}(\sigma_2)$ just plays the same action at player 2's first information set in $\mathcal{G}_{corr}$. $\mathcal{F}(\sigma_2)$ also plays the same action in G as σ_2 when player 2 is called upon to play again. Given a deterministic strategy σ_1 for player 1, if σ_1 plays a at the first step in $\mathcal{G}_{corr}$, $\mathcal{F}(\sigma_1)$ chooses uniformly at random one of the d-bit strings such that $s = a \mod \ell$ (there are at most 2 such strings), and plays the commitments strings $C_1(1^n, s_1, r_1), \ldots, C_d(1^n, s_d, r_d)$, where $r = r_1 || \ldots || r_d$ is the random string used. To play the action "open", it computes $k_i = C_2(1^n, s_i, r_i)$ and play $k_1 || \ldots || k_d$; to play "destroy", it plays $k_1 || \ldots || k_d \oplus 1$ (a string other than the right keys). Again, it is obvious how player 1 plays in G. It is easy to see that $\mathcal{F}(\vec{\sigma})$ corresponds to $\vec{\sigma}$, so UG4(a) holds. UG4(b) holds for all players trivially given these strategies.

It is also obvious that UG4(c) holds for player 1. Since the information structure it faces at $\mathcal{G}_{corr}$ and G_{corr} is essentially the same, anything it can do in $\mathcal{G}_{corr}$ can be done by a strategy in G_{corr} by just looking at the distribution of actions in histories that map to each information set.

The other players have different information structures in $\mathcal{G}_{corr}$ and G_{corr}, sinc they see the commitment strings in $\mathcal{G}_{corr}$. We discuss UG4(c) for player 2 here; the argument in the case of the others is similar (and simpler). Let σ_i for players the arguments are similar and simpler. Let σ_i for $i \neq 2$ be a strategy for player i in G_{corr}, and let $M_i = \mathcal{F}(\sigma_i)$. Let M' be an arbitrary polynomial time strategy for player 2 in $\mathcal{G}_{corr}$, and let D_1^n be the distribution over the actions of M' at player 2's first information set in G_n (which must be (independent of the commitment string); let $D_{j,w}^n$ be the distribution over the actions of M' in G given that the commitment was opened successfully,

player 1 committed to j, and player 2's first move was w; and let D^n_w be the distribution over the actions of M' in G_n if the commitment is not opened successfully and player 2's first move was w. Let σ'_n be a strategy in G_{corr} for player 2 that plays according to these distributions. We claim that $\{\phi^{G_n}_{(M_1,M',\ldots,M_c)}\}_n$ is indistinguishable from $\{\rho^{G_{corr}}_{(\sigma_1,\sigma'_n,\ldots,\sigma_c)}\}_n$.

Let $\phi^{G_n,1}_{(M_1,M',\ldots,M_c)}$ be the distribution over histories ending at the first action of player 2 when $(M_1, M', \ldots, M_c)$ is played in G_n and mapped using f_n to histories of G_{corr}, and let $\rho^{G_{corr},1}_{(\sigma_1,\sigma'_n,\ldots,\sigma_c)}$ be the distribution over partial histories ending at the first action of player 2 when $(\sigma_1, \sigma'_n, \ldots, \sigma_c)$ is played in G_{corr}. We first claim that $\{\phi^{G_n,1}_{(M_1,M',\ldots,M_c)}\}_n$ is indistinguishable from $\{\rho^{G_{corr},1}_{(\sigma_1,\sigma'_n,\ldots,\sigma_c)}\}_n$. Assume, by way of contradiction, that it is not. This can happen only if, for infinitely many n, M' plays some action a with probabilities that differ non-negligibly, depending on whether it is faced with a commitment to different strings s or s'. But that means that for infinitely many n, it can distinguish those two events with non-negligible probability. This contradicts the assumption that the commitment scheme is secure. (Note that it is easy to show that, because a single commitment is hiding, then even when d such commitments are run in parallel, no polynomial-time TM should be able to distinguish between commitments to s and s'.)

It is easy to see that this also means that the distribution over partial histories just before player 2 plays again are also indistinguishable. Now if the commitment is opened successfully, then the information structure player 2 faces in $\mathcal{G}_{corr}$ is the same as in G_{corr}, and thus the statement is obviously true. If the commitments were not opened, than by using a argument similar to that used for player 2's first action, we can argue that if the distributions over partial histories just after player 2 plays again are not indistinguishable, then again we can use that as a distinguisher for the commitment scheme. □

By Theorems 4.5 and 6.1, since $\vec{\sigma}_\mu$ (with the appropriate beliefs) is a sequential equilibrium of G_{corr}, $\mathcal{F}(\vec{\sigma}_\mu)$ is a computational sequential equilibrium of $\mathcal{G}_{corr}$.

7 Conclusion

The model introduced in this paper is a first step towards a better understanding of polynomially bounded players playing finite games. The model allows us to analyze some interesting phenomena, such as the power of state; moreover, for cryptographic protocols, it allows us to separate the cryptographic analysis from the strategic analysis.

An important next step is to refine the model so it can capture more complex cryptographic protocols. For example, some cryptographic protocols do not have a unique map between histories and actions (for example, a computationally binding commitment can map a string to both 0 or 1 depending on the key). They also might have abstract actions that are hard to compute (for instance, there might be strings that are not valid commitments at all but it might be hard to compute them), or require a few implementation steps to implement one abstract step. One possible direction is to map a history *and* the TMs' memory states into histories in the game. While this might solve some of the issues raised, it also introduces new challenges, which we intend to investigate.

References

[1] E. Ben-Sasson, A. Kalai, and E. Kalai. An approach to bounded rationality. In *Advances in Neural Information Processing Systems 19 (Proc. of NIPS 2006)*, pages 145–152. 2007.

[2] C. Borgs, J. T. Chayes, N. Immorlica, A. T. Kalai, V. S. Mirrokni, and C. H. Papadimitriou. The myth of the folk theorem. *Games and Economic Behavior*, 70(1):34–43, 2010.

[3] Y. Dodis, S. Halevi, and T. Rabin. A cryptographic solution to a game theoretic problem. In *CRYPTO 2000: 20th International Cryptology Conference*, pages 112–130. Springer-Verlag, 2000.

[4] O. Goldreich. *Foundations of Cryptography, Vol. 1*. Cambridge University Press, 2001.

[5] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[6] R. Gradwohl, N. Livne, and A. Rosen. Sequential rationality in cryptographic protocols. *ACM Trans. Econ. Comput.*, 1(1):2:1–2:38, January 2013.

[7] J. Y. Halpern and R. Pass. Algorithmic rationality: Game theory with costly computation. *Journal of Economic Theory*, 156:246–268, 2015.

[8] J. Y. Halpern, R. Pass, and D. Reichman. On the nonexistence of equilibrium in computational games. 2015.

[9] J. Y. Halpern, R. Pass, and L. Seeman. Not just an empty threat: subgame-perfect equilibrium in repeated games played by computationally bounded players. In *Proc. WINE 2014: 10th Conference on Web and Internet Economics*, pages 249–262, 2014.

[10] J.Y. Halpern, R. Pass, and L. Seeman. The truth behind the myth of the folk theorem. In *Proc. 5th Conference on Innovations in Theoretical Computer Science (ITCS '14)*, pages 543–554, 2014.

[11] P. Hubáček and S. Park. Cryptographically blinded games: leveraging players' limitations for equilibria and profit. In *Proc. 15th ACM Conference on Economics and Computation*, pages 207–208, 2014.

[12] D. M. Kreps and R. B. Wilson. Sequential equilibria. *Econometrica*, 50:863–894, 1982.

[13] H. W. Kuhn. Extensive games and the problem of information. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II*, pages 193–216. Princeton University Press, Princeton, N.J., 1953.

[14] A. Neyman. Bounded complexity justifies cooperation in finitely repeated prisoner's dilemma. *Economic Letters*, 19:227–229, 1985.

[15] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, Cambridge, Mass., 1994.

[16] A. Rubinstein. Finite automata play the repeated prisoner's dilemma. *Journal of Economic Theory*, 39:83–96, 1986.

[17] H. A. Simon. A behavioral model of rational choice. *Quarterly Journal of Economics*, 49:99–118, 1955.

[18] A. Urbano and J. E. Vila. Computationally restricted unmediated talk under incomplete information. *Economic Theory*, 23(2):283–320, 2004.

[19] P. C. Wichardt. Existence of Nash equilibria in finite extensive form games with imperfect recall: A counterexample. *Games and Economic Behavior*, 63(1):366–369, 2008.