Perfect Concrete Implementation of Arbitrary Mechanisms

(A quick summary of joint work with Sergei Izmalkov and Matt Lepinski)
[Extended Abstract]

Silvio Micali
MIT
silvio@csail.mit.edu

1. THE PROBLEM

Privacy and trust affect our everyday thinking and, in particular, the way we approach a concrete game. Accordingly, we hope that a rigorous treatment of privacy and trust will become integral part of mechanism design. As of now, the field has been very successful in finding many ingenious mechanisms as solutions to a variety of problems. But these mechanisms are *theoretical constructions* and not enough attention has been devoted to their concrete implementation. Indeed, It should be appreciated that the outcome function of a simple normal-form mechanism does not spontaneously evaluate itself on the "messages" that the players have selected in "their own minds." To be practically useful in a real strategic setting, any mechanism $\mathcal{M}$, whether of normal or extensive form, must be *concretely implemented.* But then, in such concrete implementations, issues of privacy and trust may arise so as to *undermine* the valuable theoretical properties of $\mathcal{M}$.

For instance, consider concretely implementing a second-price auction using a mediator M, who privately receives the players' bids, and then announces the winner of the good and the price he has to pay. Since the players have no way of verifying that such announcement indeed consists of the second-highest bid, M needs to be *trusted.* After all, nothing prevents M from boosting the auction's revenue by manufacturing out of thin air a second-highest bid artificially close to the highest one. A player worrying about this possibility would thus be tempted to "underbid," putting at risk the valuable dominant-strategy truthfulness of the second-price mechanism.

Since universally and completely trusted mediators are hard to come by, one may consider concretely implementing a second-price auction by asking the players to submit their bids sealed into opaque *envelopes.* After all envelopes have been collected in public view, they are publicly opened so that everyone present can verify that winner and price have been derived correctly. This concrete implementation, however, violates the privacy of all bids and thus may alter the way in which privacy-valuing players behave. On one hand, a privacy-valuing player receives a negative utility when his valuation is publicly revealed; on the other, the second-price mechanism gives him incentives to reveal his valuation truthfully. What the result of these opposing forces will be is far from clear. Again, therefore, the dominant-strategy truthfulness of the abstract second-price may not hold in practice.

On the basis of these examples, we wish to pose and tackle the following problem:

> *Is there a way to implement concretely an abstract mechanism so as to preserve its strategic properties without relying on a trusted mediator or violating the players' privacy?*

More precisely, our goals are (1) defining what a "perfect" solution to the above problem should be, and (2) finding one such solution for any possible mechanism. Note that both goals are model-sensitive. Indeed, the above problem is provably unsolvable in most models —in particular, in the model of Brandt and Sandholm (2004). But this should not affect our resolve. Indeed, we are not interested in solving the above problem in *all* models, but in finding reasonable models in which a solution always exists.

We insist on perfect solutions because we believe that this is the right way to start a rigorous investigation of privacy and trust in mechanism design. Only after gaining a clearer understanding of what is in principle available to us can we meaningfully discuss what "compromises" are worth making.

2. AN OLD AND IMPERFECT SOLUTION

A non-perfect solution to our problem could be derived from an older general result of Goldreich, Micali, and Wigderson (1987). This result, referred to in the cryptographic literature as *general and secure multi-party computation* can be informally stated as follows, assuming for simplicity that n, the number of players, is greater than 2. For each finite function $f : X_1 \times \cdots \times X_n \to Y$, there exists a communication protocol P that, if honestly followed by the majority of players, enables the computation of $y = f(x_1, \ldots, x_n)$ with the same correctness and privacy as when each player i privately gave his own secret input $x_i \in X_i$ to a trusted mediator, who then evaluates f on the received inputs and then announces the result. A bit more precisely, P is such that no subset s comprising $< n/2$ of the players, even if capable of perfectly coordinating a joint and arbitrary deviation from P, can alter the correctness of the result or learn any information about the input subprofile x_{-s} that is additional to that implicitly revealed by y itself.

The relevance of this result to the concrete implementation of any finite normal-form mechanism $\mathcal{M}$ should be quite clear. In $\mathcal{M}$, let X_i be the set of possible strategies (i.e., "messages") of player i, Y the set of possible outcomes, f the outcome function, and x an equilibrium. Then, to implement concretely this equilibrium, rather than having

Copyright is held by the author/owner(s).
BQGT'10 May 14-16, 2010 Newport Beach, California USA
ACM 978-1-60558-919-0.

each player i confide x_i to a mediator, the players may instead execute a protocol P for securely evaluating f. In such an implementation the set of possible strategies of a player i greatly increases. Indeed, they include not only all possible strings in X_i, but also all possible ways for i to behave in the communication protocol P —that is, his prescribed communication strategy as well as all his possible ways to deviate from it. Nonetheless the new, concrete, and unmediated game continues to have an equilibrium corresponding and indeed payoff-equivalent to the original equilibrium x: namely, the equilibrium in which each player i chooses string x_i and honestly follows σ_i, his prescribed communication strategy for securely evaluating f. Very roughly, one can argue that the "strategy profile" (x, σ) is an equilibrium of this new game goes as follows. Assume that a player i deviates from by choosing a string x'_i and a communication strategy σ'_i, while any other player j sticks to x_j and σ_j. The, because $n > 2$ and thus $1 < n/2$, in this execution of the secure computation protocol only a minority of the players has deviated. Accordingly the communication protocol by definition correctly and privately computes $y' = f(x_{-i}, x'_i)$, that is, an outcome providing i a utility no greater greater than that which he would get by choosing x_i and honestly communicating according to σ_i. In the latter case in fact, the outcome would be $y = f(x)$, and x is an equilibrium.

Of course, implementing a trusted mediator by a secure evaluation of the outcome function f requires some additional effort, both in computation and communication. But such additional effort is not unfeasible.[1] The real limitation of this approach is that it does not go much beyond preserving all equilibria of the original mechanism $\mathcal{M}$. While this is a non-trivial achievement, we should demand much more from a perfect concrete implementation of $\mathcal{M}$. In particular, we should demand that the power of coalitions of players, whatever it may be, is preserved too. The problem of providing excessive power to coalitions has not received much attention in mechanism design (with some notable exceptions, see Eliaz (2002) and Laffont and Martimort (2000)), but is crucial for concrete implementation. Let us explain.

The only guarantee offered by an equilibrium σ is that no individual player i has any incentive to deviate from σ_i. But all bets are off if two or more players jointly deviate from their strategies in σ. In the original mechanism $\mathcal{M}$, this may not be a problem. For instance, because the players do not have sufficient *means* or sufficient *incentives* to collude. But the situation changes dramatically when $\mathcal{M}$ is implemented by a secure evaluation of the outcome function f as discussed. This is so because while any minority s of the players is *powerless,* any majority S of the players is *omnipowerful.* This means that, by coordinating their strategies in the secure computation protocol, the members of S can (1) learn everything about x_{-S}, the other players' inputs, and (2) force f's output to be any $y \in Y$ they choose, even one "incompatible" with the other players' inputs.

Let us consider the following example, where $n = 5$, the strategies available in $\mathcal{M}$ to each of the 5 players are 0 and 1, the outcomes are A, B, and C, the players know each other's utilities, and the outcome function f and the utility profiles for the various outcomes are as follows:

- $f(0,0,0,0,0) = A$, and $u(A) = (10, 10, 10, 10, 10)$;
- $f(1,1,1,1,1) = B$ and $u(B) = (10^{10}, 10^{10}, 10^{10}, -\infty, -\infty)$; and
- for any other strategy profile x, $f(x) = C$ and $u(C) = (-\infty, -\infty, -\infty, -\infty, -\infty)$.

Then $(0,0,0,0,0)$ is the only equilibrium composed of weakly undominated strategies, and thus it is reasonable to expect the outcome resulting from a play of $\mathcal{M}$ to be A. Assume now that the players decide to implement $\mathcal{M}$ by running a protocol P securely evaluating f. Then, players 1, 2, and 3 form a majority, and therefore can control the outcome at will and with total impunity. In such a concrete implementation, therefore, we might expect B to be the outcome.

We thus believe that, for any set of players S, preserving the strategic and privacy power available to S is a good way to prevent that the players in S, while unwilling or unable to collude in an abstract mechanism, might do so in its concrete implementations.

3. A PERFECT SOLUTION FOR NORMAL-FORM MECHANISMS

The notion of a perfect concrete implementation of a normal-form mechanism, was provided in Izmalkov, Lepinski and Micali (2005). Informally, for a normal-form (and thus "abstract") mechanism $\mathcal{M}$ to be perfectly implemented by another ("concrete") mechanism $\mathcal{M}'$ the following three properties must hold:

- *Strategy Equivalence:* For each player i there exists a bijection ψ_i between i's strategies in $\mathcal{M}$ and his strategies in $\mathcal{M}'$ such that for each profile of strategies σ in $\mathcal{M}$ we have

$$\begin{aligned}\mathcal{M}(\sigma) &= \mathcal{M}(\sigma_1, \ldots, \sigma_n) = \mathcal{M}'(\psi_1(\sigma_1), \ldots, \psi_n(\sigma_n)) \\ &= \mathcal{M}'(\psi(\sigma))\end{aligned}$$

  where the above equalities are among distributions if $\mathcal{M}$ and $\mathcal{M}'$ are probabilistic.

- *Privacy Equivalence:* For any strategy profile σ of $\mathcal{M}$, and any subset S of the players, the information learnable by the players in S in mechanism $\mathcal{M}$ under σ coincides with that learnable by the same players in $\mathcal{M}'$ under $\psi(\sigma)$.

- *Complexity Equivalence:* The number of elementary operations needed to execute $\mathcal{M}'$ under $\psi(\sigma)$ is essentially equal to those required to execute $\mathcal{M}$ under σ.[2]

It is worth to point out that while $\mathcal{M}$ is normal-form, $\mathcal{M}'$ needs not to be. Yet, the above properties guarantee that the players are perfectly indifferent between playing $\mathcal{M}$ or $\mathcal{M}'$, not only from a strategic view point, but also from privacy or computational perspective.

Notice that, in particular, perfect implementation implies that a strategy profile σ is an equilibrium of the game G corresponding to $\mathcal{M}$ if and only if $\psi(\sigma)$ is an equilibrium of the game G' corresponding to $\mathcal{M}'$. It also implies that σ

[1] That, is, the time it takes to securely compute any function f is upperbounded by a fixed polynomial of the time required to compute f itself, without any privacy or robustness constraints. This said, to concretely implement a specific mechanism, and thus a specific outcome function f, one is computationally better off by designing an ad hoc secure protocol for evaluating the specific f at hand rather than invoking a general secure computation protocol.

[2] One can actually prove any $\mathcal{M}$ has a perfect implementation $\mathcal{M}'$ for which the said number of elementary operations differ by at most a multiplicative factor of 128.

and σ' are payoff-equivalent. More importantly, it implies that the members of any subset S of players, no matter what the cardinality of S may be, always have the same strategic opportunities as well as the same knowledge about the strategies adopted by the players in $-S$ in G as in G'. In particular, therefore, the members of S have the same capabilities and incentives to collude in G' as they have in G. If they have no such capabilities in G, they do not have in G'. Else, they have as much to gain from colluding in G as they have from colluding in G'.

Beyond putting forward the notion of a perfect implementation, the contribution of Izmalkov, Lepinski and Micali (2005) consists of proving that every normal-form mechanism $\mathcal{M}$ has such an implementation $\mathcal{M}'$ using ballots and a ballot randomizer —that is, the same traditional machinery used from time immemorial to implement a private election, or a fair lottery.

4. A PERFECT SOLUTION FOR ARBITRARY MECHANISMS

In a more recent paper, Izmalkov, Lepinki, and Micali (2008), the notion of a perfect implementation is extended from normal-form to quite general mechanisms of extensive forms. Such mechanisms can be conceptualized as repeated interactions between the players and a trusted mediator T, in which each party can keep a local state. The interaction proceeds in stages. Essentially, in a stage each player secretly sends a message to T, who then secretly computes a pre-specified probabilistic function of the received messages so as to determine (1) a separate secret response for each player, that he then privately sends to that player, and possibly (2) a common string that he publicly announces. The players now may select new messages for T, and so on. The mechanism can generate a single final outcome, or a sequence of individual outcomes, one for each stage.

At a high level, a perfect implementation of such a mechanism $\mathcal{M}$ again consists of a concrete mechanism $\mathcal{M}'$ equivalent to $\mathcal{M}$ from the perspective of strategy, privacy, and complexity. The main problem consists of showing that such an implementation indeed exists for each possible $\mathcal{M}$. The solution, also provided in the same paper, is somewhat more complex than in the case of normal-form mechanisms, due to several new demands imposed by the interactive nature of mechanism $\mathcal{M}$: namely, $\mathcal{M}'$ should not *introduce* any "entropy not present in the original $\mathcal{M}$." Let us explain.

Notice that any stage of $\mathcal{M}$ can, in particular, consist of a mechanism of some normal-form game G. Indeed, the players may privately send the trusted mediator T their selected strategies in G, and then T may announce G's outcome. Therefore, assume that playing such a profile of strategies is indeed what happens in a stage $s > 1$ of $\mathcal{M}$. As it is well known, if a common random signal were publicly available, then some additional strategic options would be available to the players of G. In this case, in fact, the players might also be able to "convexify" the original Nash equilibria of G. Thus if the first $s-1$ stages of $\mathcal{M}$ do not cause any random string to become common knowledge to the players, then neither should a perfect implementation $\mathcal{M}'$ of $\mathcal{M}$, else the strategic analysis and play of the game generated by $\mathcal{M}$ in stage s would be different from the one generated by $\mathcal{M}'$. Assume now that the first stage of $\mathcal{M}$ consists of conducting a secret referendum between option 0 and option 1: that is, each player i privately sends the trusted mediator T a bit b_i, and then T announces the tally $t = \sum_i b_i$. It would seem that this elementary stage could be trivially simulated in $\mathcal{M}'$ via ballots and a ballot randomizer as follows: each player i seals his vote b_i into a ballot, all ballots are inserted in the randomizer, randomized, and then publicly opened so as to enable anyone to compute the tally t without betraying any information about the original votes that is not implicit in t itself. However, while securely computing t, this way of proceeding also causes some randomness to become common knowledge. Indeed, the players do not just learn t, but also a random element in the set, of cardinality n-choose-t, consisting of all n-bit sequences with t 1's. This "common" randomness therefore may affect the strategic way of the players to subsequently play G. Thus, in order to satisfy strategy equivalence, $\mathcal{M}'$ must use the ballots and the ballot randomizer in a way that guarantees that, once all ballots are publicly opened, not one of the possible n-choose-t bit sequences with t 1's will be revealed, but a fixed one: for instance, the one consisting consisting of t 1's followed by $n-t$ 0's. This may sound somewhat counter-intuitive, and indeed some effort is required to achieve it in a way "verifiable" by all players.

5. A SIMPLER BUT NON-TRIVIAL SPECIAL APPLICATION: CORRELATED EQUILIBRIUM

As put forward by Aumann (1974), the notion of correlated equilibrium extends the notion of NASH equilibrium and enables rational players to reach payoffs higher than those in any Nash equilibrium for some games. However, to achieve these payoffs rational players have to rely on a trusted mediator T to implement the specific correlated equilibrium: to sample separate signals according to a special joint distribution, and then privately give each signal to the proper player. Perfectly implementing such a mediator T poses various strategic, computational, trust, and privacy problems. Yet, such a T is just a specific example of a general mechanism of extensive form, and thus it is concretely implementable by the general construction in Izmalkov, Lepinski and Micali (2008).

To be sure, this concrete implementation is the last one in a long series, but also the first one to be "perfect" in the sketched technical sense. Indeed, much effort has been devoted to achieve correlated-equilibrium payoffs by adding a pre-play communication stage to the main game: see in particular, Bárány (1992), Forges (1990), Ben-Porath (1998), Dodis, Halevi and Rabin (2000), Urbano and Vila (2002), Aumann and Hart (2003), Ben-Porath (2003), Gerardi (2004), Krishna (2006), Gerardi and Myerson (2007). These specific solutions, however, trade conceptual simplicity for computational efficiency and are exponentially far from being complexity-equivalent to achieving correlated equilibrium with a trusted mediator T.[3] In addition, neither of them satisfies strategy equivalence. To achieve a correlated equilibrium E in a normal-form game G, they construct an "extended

[3]That is, there exists 2-player, 2-strategy, normal-form games G with k-bit utilities for which a trusted mediator can put the players in correlated equilibrium in a number of elementary operations linear in k, while the above methods require time, communication, and other resources (e.g., number of envelopes) exponential in k.

game" G' having an equilibrium that is payoff-equivalent to E. But such a G' also has additional equilibria that do not have any counter-part in G. Accordingly, a play of G may be vastly different from a play of G'. The implementations closer to achieving strategy equivalence are those of Ben-Porath (1998) and Krishna (2006). But they too miss strategy equivalence in its purest form. That is, while in a game G with two correlated equilibria E and E' a trusted mediator T might be able to enable the players to reach E but not E', the latter two implementations cannot separate reaching E from reaching E'. That is, whenever they enable the players to reach E, they automatically enable them to reach E' as well. While this may not be bad from the players' perspective, it is unacceptable from a mechanism design perspective. If, somehow, society needed to obtain an outcome distributed according to E, then enabling the players to obtain instead an outcome distributed according to E', as they indeed prefer, would be a bad idea.

In any case, if one wanted to use a concrete implementation of correlated equilibrium E as a component of the concrete implementation of a larger and more general extensive-form mechanism, he would need to be able to provide a perfect solution to the simpler problem of reaching equilibrium E and E alone. Let us now explain why this may be non trivial.

Let G be the following two-player game,

	A	B	C	D
A	$9,6$	$-100,-100$	$-100,-100$	$-100,-100$
B	$-100,-100$	$6,9$	$-100,-100$	$-100,-100$
C	$-100,-100$	$-100,-100$	$4,4$	$1,5$
D	$-100,-100$	$-100,-100$	$5,1$	$0,0$

let E be the correlated equilibrium of G which assigns probability $1/5$ to each of outcomes (A,A), (B,B), (C,C), (C,D) and (D,C), and consider the problem of concretely implementing E *perfectly*. That is, as when a trusted mediator T selects one of these 5 outcomes uniformly, privately tells each player his strategy component in the selected outcome, and promises not to reveal to either player any additional information about the strategy of his opponent.

To illustrate the difficulty of doing so, consider the following simpler variant of pre-play implementations of Ben-Porath (1998) and Krishna (2006). The player have access to *envelopes* indistinguishable from each other, and to larger *super-envelopes*, also indistinguishable from each other. They publicly construct 5 super-envelopes, each containing two distinct envelopes, one marked "player 1" and the other marked "player 2". The envelopes inside the first super-envelope respectively contain "A" and "A", those inside the second super-envelope contain "B" and "B", those inside the third "C" and "C", those inside the forth "C" and "D", and those inside the fifth "D" and "C". Then the players place all 5 super-envelopes in an opaque bag, verified by both of them to be initially empty, and play the following protocol for picking a super-envelope e at random. First, player 1 privately and randomly permutes the 5 super-envelopes and hands the bag to player 2. Then, player 2 privately takes out a random super-envelope from the bag. The bag is now destroyed with its remaining inner ballots, while the super-envelope taken out by player 2 is publicly opened, so that the former inner envelope labeled "player 1" is taken by player 1 and that labeled "player 2" by player 2. Each of two players now privately opens his own envelope, learns his own recommendation, and proceeds to play the normal-form game G. As it is clear, the game G' consisting of this overall procedure has E as one of its equilibria. However, *G' fails to match the set of equilibrium outcomes reachable in G with the assistance of the angel generating a profile of private recommendations distributed according to E (and E only).*

Notice that the two players actually prefer to E the following alternative correlated equilibrium E': (A,A) with probability $1/2$ and (B,B) with probability $1/2$. Notice too that, when T puts the two players in equilibrium E —that is, when he provides them with E-correlated recommendations,— the players cannot use these recommendations in order to achieve E'. That is, they have no way to turn their E-correlated recommendations into E'-correlated ones. However, in the above constructed extended game G', they could also easily achieve correlated equilibrium E' without anyone else noticing anything wrong. Namely, consider the following profile σ' of strategies in G':

σ'_1: protected by the privacy given him by the bag, rather than randomly permuting all 5 super-envelopes, player 1 first randomizes just the first two super-envelopes —that is those respectively containing the recommendations (A,A) and (B,B)— keeping both envelopes in positions 1 and 2, and then randomizes the other 3 super-envelopes, keeping them in positions 3, 4, and 5.

σ'_2: When his turn comes, player 2 randomly selects one of the first two super-envelopes.

Now notice that (1) σ' implements E' rather than E; (2) σ' is an equilibrium of G'; and (3) no external observer can tell during the execution of σ' which strategy the players are running.[4]

Moreover, selecting equilibrium σ' does not require any special coordination or risk in the following sense. When player 1 deviates from his strategy in equilibrium σ to σ'_1 while player 2 sticks to strategy σ_2, the payoff of player 1 remains the same (and the same holds reversing the roles of players 1 and 2).[5]

Mutatis mutandis, the same "additional-equilibrium problem" described above, together with the already discussed equilibrium-selection problems, arise when implementing E via the protocol of Dodis et al. (2000). The latter implementation essentially consists of a custom-constructed secure computation protocol for the specific probabilistic function of correlated equilibrium. Alternative solutions based

[4]Nor can he obtain ex post any proof that the players have executed σ' instead of the intended equilibrium σ. After all, the permutations in the support of σ' are all legitimate ones.

[5]Of course, one could consider straightforwardly using a ballot randomizer in order to pick a random super-envelope e in the above variant of the Ben-Porath/Krishna protocol. But then the modified protocol would be not satisfy "complexity equivalence." Consider a correlated equilibrium in which outcome (A,A) is selected with probability $1/10$ and outcome (B,B) with probability $9/10$. Then this approach would implement this equilibrium by preparing 10 super-envelopes: one containing a pair of envelopes with respective contents "A" and "A", and 9 containing a pair of envelopes having "B" and "B" as their respective contents. And if a correlated equilibrium would require (A,A) to be selected with probability 2^{-k}, then this approach would require at least 2^k envelopes, despite the fact the a trusted mediator can easily select (A,A) with the right odds by just flipping k coins.

on secure computation may also suffer of a similar, if not identical, problem. The point is that for the players to be able to implement E' instead all that is needed is a single, common, random bit b. And in all traditional two-party protocols for secure computation the players use "coin flipping" as a subroutine.

6. CONCLUSIONS

Designing a mechanism $\mathcal{M}$ without worrying about its concrete implementation is designing an abstraction. Abstractions simplify our lives by enabling us to focus on crucial aspects without worrying about details. But should there be no way to concretely approximate these abstractions, their usefulness would be greatly challenged. By proving that any mechanism can be concretely implemented in a perfect way, we make it possible for a designer to ignore issues of privacy and trust when implementing her work. Done this, we can now explore less perfect ways, but also more practical ways, to achieve at least in part what has been achieved here.

7. REFERENCES

Aumann, R. J.: 1974, Subjectivity and correlation in randomized strategies, *J. Math. Econ.*, **1**, 67–96.

Aumann, R. J. and Hart, S.: 2003, Long cheap talk, *Econometrica* **71**(6), 1619–1660.

Bárány, I.: 1992, Fair distribution protocols or how the players replace Fortune, *Mathematics of Operations Research* **17**, 329–340.

Ben-Porath, E.: 1998, Correlation without mediation: Expanding the set of equilibrium outcomes by cheap pre-play procedures, *Journal of Economic Theory* **80**, 108–122.

Ben-Porath, E.: 2003, Cheap talk in games with incomplete information, *Journal of Economic Theory* **108**(1), 45–71.

Bolton, P. and Dewatripont, M.: 2005, *Contract Theory*, MIT Press, Cambridge, Massachusetts.

Brandt, F. and Sandholm, T.: 2004, (Im)possibility of unconditionally privacy-preserving auctions, *Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, IEEE, pp. 810–817.

Dodis, Y., Halevi, S. and Rabin, T.: 2000, A cryptographic solution to a game theoretic problem, *Advances in Cryptology — CRYPTO 2000, LNCS*, Vol. 1880, Springer-Verlag, pp. 112–130.

Eliaz, K.: Fault-tolerant implementation, Review of Economic Studies **69**, 589-610.

Laffont, J. J. and Martimort, D.: Mechanism design with collusion and correlation, Econometrica, **68**, pp. 309-342.

Forges, F. M.: 1990, Universal mechanisms, *Econometrica* **58**, 1341–1364.

Gerardi, D.: 2004, Unmediated communication in games with complete and incomplete information, *Journal of Economic Theory* **114**(1), 104–131.

Gerardi, D. and Myerson, R. B.: 2007, Sequential equilibria in bayesian games with communication, *Games and Economic Behavior* **60**(1), 104–134.

Goldreich, O., Micali, S. and Wigderson, A.: 1987, How to play any mental game, *Proceedings of the 19th Symposium on Theory of Computing*, ACM, pp. 218–229.
An additional version, *How to Solve Any Protocol Problem*, is available at `http://people.csail.mit.edu/silvio/SelectedScientificPapers/SecureProtocols`.

Izmalkov, S., Lepinski, M. and Micali, S.: 2005, Rational secure computation and ideal mechanism design, *Proceedings of the 46th Symposium on Foundations of Computer Science*, IEEE, pp. 585–594.
The full version, called *Perfect Implementation*, has been accepted by Games and Economic Behavior, subjet to revisions. It is available at `http://people.csail.mit.edu/silvio/SelectedScientificPapers/MechanismDesign`.

Izmalkov, S., Lepinski, M. and Micali, S.: 2008, Verifiably Secure Devices, *Proceedings of the 5th Theory of Cryptography Conference*, Springer, pp. 273-301.

Krishna, R. V.: 2006, Communication in games of incomplete information: Two players, *Journal of Economic Theory*. forthcoming.

Lepinski, M., Micali, S., Peikert, C. and Shelat, A.: 2004, Completely fair sfe and coalition-safe cheap talk, *Proceedings of the 23rd annual Symposium on Principles of distributed computing*, ACM, pp. 1–10.

Lindell, Y. and B. Pinkas: 2004, A Proof of Yao's Protocol for Secure Two-Party Computation, available at: `http://www.cs.biu.ac.il/~lindell/abstracts/yao_abs.html`.

Naor, M., Pinkas, B. and Sumner, R.: 1999, Privacy preserving auctions and mechanism design, *Proceedings of the 1st conference on Electronic Commerce*, ACM.

Urbano, A. and Vila, J. E.: 2002, Computational complexity and communication: Coordination in two-player games, *Econometrica* **70**(5), 1893–1927.

Yao, A.: 1986, Protocol for secure two-party computation, never published. The result is presented in Lindell and Pinkas (2004).