Social Choice and Preference Protection

Towards Fully Private Mechanism Design

Felix Brandt
Computer Science Department
Technical University of Munich, Germany
brandtf@cs.tum.edu

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Economics, Security

With the growing number of electronic markets on the net, there comes a growing demand for protection of privacy in electronic mechanisms. Instead of simply having to rely on the trustworthiness of system operators, cryptography provides the tools to ensure privacy in other ways. This extended abstract intends to bring the fields of mechanism design and secure multiparty computation together by generalizing some results already obtained in the area of cryptographic auction protocols [1]. We point out that secure social choice mechanisms can be constructed by distributing the mechanism computation on the participants themselves. This is achieved by using multiparty computation that is not based on any trusted fraction (threshold) assumptions. We show that the main reason for thresholds in the cryptographic model is a robustness requirement that can be loosened in our case. As a consequence, the correct and private execution of mechanisms can be guaranteed in the absence of trusted third-parties. In [2] the first (and, to the best of our knowledge, only) scheme to privately evaluate mechanisms has been proposed. Besides some similarities, their approach substantially differs from ours as they use two third-parties that are assumed not to collude.

The aggregation of conflicting preferences in a group of agents is one of the central topics of economics and multiagent systems. Two major problems have been considered in this context so far.

The **social choice problem** is to find a function that "*fairly*" aggregates conflicting preferences.

The **mechanism design problem** is to construct mechanisms that urge self-interested agents to reveal preferences *truthfully*.

Classically, the existence of a central institution that receives all preferences and resolves the mechanism is assumed. However, neither the correctness of the result nor the privacy of the individual inputs can be guaranteed. Especially, incentive-compatible mechanisms might deter agents from participating as they require the submission of true valuations. Confidentiality of these valuations is essential for future negotiations and its revelation can be disastrous. We therefore introduce the "**preference protection problem**", which is the problem to enable the correct execution of a mechanism without trusted third-parties while maintaining privacy, i.e. preventing agents from learning preferences of other participants.

We suggest that a subfield of cryptography called "secure multiparty computation" is the key to solve the preference protection problem. Similar to the *implementation* of social choice functions in mechanisms, our new view on public choice adds another level to the model by introducing the *emulation* of mechanisms by cryptographic protocols. We say that a protocol is *fully private* if it is secure despite any collusion of participants.

Secure multiparty computation (MPC) deals with protocols that allow n parties to jointly compute a function $f(x_1, x_2, \ldots, x_n) = (y_1, y_2, \ldots, y_n)$ on their individual private inputs x_i, so that agent i only learns y_i but nothing else. The common model defines passive adversaries (or "eavesdropping adversaries") as agents that follow the protocol but try to derive additional information. Active adversaries, on the other hand, try to violate privacy and correctness by every means including the sending of faulty messages. Furthermore, there are two classes of protocols. The security of *computational* protocols is based on complexity assumptions, i.e., they are only safe against computationally polynomially bounded adversaries. *Unconditional* (or information-theoretic) protocols provide perfect security given that agents can communicate via private channels. Typically, secure MPC is accomplished by having each agent distribute shares of his individual input on the other participants. This has to be carried out in conjunction with a commitment scheme, so that agents can verify the correctness of shares. This primitive is called "verifiable secret sharing". In the following, the participants verifiably evaluate a Boolean circuit representing function $f(\cdot)$ with their shares as inputs and new shares as outputs. When the evaluation of the circuit is finished, agents broadcast their resulting shares and reconstruct the final result.

Unconditionally secure MPC is possible if there are not more than $\lfloor \frac{n-1}{3} \rfloor$ active adversaries or, when only consid-

EC'03, June 9–12, 2003, San Diego, California, USA.
ACM 1-58113-679-X/03/0006.

ering passive adversaries, not more than $\lfloor \frac{n-1}{2} \rfloor$ passive adversaries. The reasons for thresholds in unconditional multiparty computation are:

1. Robustness (threshold: $\frac{n}{2}$)
2. Feasibility of secure broadcasting (threshold: $\frac{n}{3}$)
3. Feasibility of verifiable secret sharing (threshold: $\frac{n}{3}$)
4. Feasibility of secure Boolean OR (threshold: $\frac{n}{2}$)

Now, let us try to make weak assumptions that might allow unconditional secure MPC without thresholds. First of all, **robustness** against active adversaries in MPC is defined to allow correct completion of the computation even if active adversaries do not follow the protocol. If a majority of the participants is assumed to be cooperating, the shares can be distributed in a way that allows any majority of agents to reconstruct the original values, including the inputs of malicious participants. This ensures robustness so that no minority quitting the protocol can prevent the correct execution of the protocol. When presuming that active adversaries can be detected and "kicked out", *including* their inputs, this leads to a weaker, but for our purpose sufficient, notion of robustness. We say a protocol is *weakly robust* if the correct computation of function $f(X)$ of inputs supplied by non-adversaries $X \subseteq \{x_1, x_2, \ldots, x_n\}$ can always be completed. Of course, this only makes sense if $f(\cdot)$ is defined for any number of inputs up to n. A weakly robust protocol terminates after at most $n-1$ iterations (excluded adversaries do *not* learn any information). If participation in a mechanism is voluntary, the outcome function of a mechanism is defined for an arbitrary number of inputs n. Public verifiability of the protocol is sufficient to provide weak robustness and verifiability can be easily achieved by using zero-knowledge proofs (like in [1]). Unfortunately, when abandoning strong robustness, we also lose "fairness": Typically, in the end of a protocol run, each participant holds a share of the result. As simultaneous publication of these shares is impossible, a malicious agent might quit the protocol after having learned the result but before others were able to learn it. There are various techniques to approximate fairness by gradually releasing parts of the secrets to be swapped. Another possibility is to introduce a third-party that publishes the outcome after it received all shares. This third-party does not learn confidential information. It is only assumed not to leave the protocol prematurely. We learned that in auctions with a single seller, it is practical to assign this role to the seller [1].

Broadcasting, i.e. sending one message to all other agents, is not generally possible (without a trusted third-party) because it has to be guaranteed that all agents receive the *same* message. It has been shown that reliable broadcasting can be achieved in the presence of $\lfloor \frac{n-1}{3} \rfloor$ (active) adversaries in the unconditional case. Providing a secure broadcast channel can eliminate this threshold.

Without making any assumptions, **verifiable secret sharing** can only be accomplished when more than two thirds of the participants are honest. As shown in [3] verifiable secret sharing can provide unconditional security of either the shares' correctness or unconditional privacy of the secret, but not both, without any threshold. The latter seems much more practical since it means that the individuals' preferences can *never* be revealed. A malicious agent, however, can manipulate the protocol by applying super-polynomial computational power *during* the protocol.

It has been proven that the secure computation of essential **boolean gates** like OR and AND is impossible in the unconditional model (if more than half of the (passive) participants can pool their knowledge). Unfortunately, this threshold cannot be removed.

PROPOSITION 1 (UNCONDITIONAL MECHANISM EMULATION) It is impossible to emulate arbitrary mechanisms by fully private protocols in the unconditional model, even when assuming weak robustness, providing a broadcast channel, and accepting the possibility of manipulation by computationally unbounded cheaters.

Please note that like the impossibility of strategy-proof implementations for *general* preferences in the Gibbard-Satterthwaite Theorem, Proposition 1 only states the impossibility of a general mapping from mechanisms to protocols, i.e., most mechanisms cannot be emulated by fully private protocols. However, there are some primitive mechanisms that can be emulated under the assumptions of Proposition 1, e.g., the sum of n input values can be computed fully private, weakly robust in the unconditional model if we accept the (theoretical) possibility of manipulations by computationally unbounded participants. Some MPC protocols work on finite fields instead of binary values. In these arithmetic protocols, addition (and thus XOR and NOT gates) are feasible while multiplication of shares is impossible (multiplication could be used to build OR or AND gates). Another example for an unconditional, fully private protocol is the Dutch auction. This protocol emulates the first-price sealed-bid auction without any intractability assumptions.

When allowing intractability assumptions, most of the reasons why unconditional MPC is impossible without threshold assumptions can be removed. The classic systems are based on the existence of trapdoor one-way permutations like the problem of factoring large composite numbers, or the decisional Diffie-Hellman problem (related to the difficulty of computing discrete logarithms). All the assumptions needed in the computational model can be reduced to the existence of "oblivious transfer" which can be achieved by noisy channels, trapdoor functions, or quantum channels. In this setting, primitives like broadcasting and verifiable secret sharing [3], and the secure computation of OR gates are feasible without threshold assumptions. With the aid of our notion of weak robustness, this yields the following proposition.

PROPOSITION 2 (COMPUTATIONAL MECHANISM EMULATION) Any mechanism can be emulated by a fully private, weakly robust protocol in the computational model.

The naive emulation of a mechanism can be extremely inefficient because general cryptographic multiparty computation protocols work on single bits and have excessive complexities. Therefore, the design of efficient, specialized protocols remains a problem.

[1] F. Brandt. Fully private auctions in a constant number of rounds. In *Proc. of the 7th International Conference on Financial Cryptography*, Springer LNCS, 2003

[2] M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and mechanism design. In *Proc. of the 1st ACM Conference on Electronic Commerce*, pages 129–139, 1999.

[3] T. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Proc. of the 11th Crypto*, pages 129–140, Springer LNCS 576, 1991.