Cryptography and Mechanism Design

Moni Naor*
Dept. of Computer Science and Applied Math
Weizmann Institute of Science
Rehovot 76100, Israel
naor@wisdom.weizmann.ac.il

1 Introduction

Mechanism Design is the algorithmic component of Game Theory, the synthesis of protocols for selfish parties to achieve certain properties. A protocol is a method to aggregate the preferences of the parties in order to decide on some "social choice," where typical examples include: deciding whether a community should build a bridge, how to route packets in a network and deciding who wins an auction. Each party has a utility function which expresses how much it values each possible outcome of the protocol. The goal is to design a protocol where the winning strategies achieve the social choice. Recently Mechanism Design has received attention by computer scientists in light of the above applications, see [19, 20].

An important result in Mechanism Design is the *Revelation Principle* that essentially states that if there is a game with dominating strategies achieving a given social choice function, then there is a mechanism where the parties simply report their true utility functions. The resulting protocol is therefore: each party send information about its utility function to a center. The center then decides on the outcome of the protocol based on the parties' reports. It is often assumed that the center can be trusted by the parties, but this might not always be true, especially in an Internet environment. The revelation principle might not be applicable if the center is corrupt and misuses the truthful bids it receives. Privacy is therefore essential in order to ensure the center's credibility. This problem was stated by Varian [23] as follows: *"Even if current information can be safeguarded, records of past behavior can be extremely valuable, since historical data can be used to estimate the willingness to pay. What should be the appropriate technological and social safeguards to deal with this problem?"*

Cryptography deals with preserving the secrecy and integrity of data in computer and communication systems. An exciting topic of cryptographic research in the last two decades is *secure function evaluation* (see [6] for an introduction). For any function $f(x_1, x_2, \ldots, x_n)$ it is possible *in principle* to construct a protocol that allows a group of n parties, where party i has as its private input α_i, to jointly evaluate $f(\alpha_1, \alpha_2, \ldots, \alpha_n)$. Following the protocol the parties learn $f(\alpha_1, \alpha_2, \ldots, \alpha_n)$ but no party i can learn about the other inputs $\{\alpha_j\}_{j \neq i}$ more than can be computed from α_i and $f(\alpha_1, \alpha_2, \ldots \alpha_n)$.

Given that any mechanism can be considered as an evaluation of a function of the utility function, it is tempting to try and use such a secure evaluation protocol in order to eliminate the

*Work performed while at the IBM Almaden Research Center and Stanford University.

trusted center assumption. However there are number of issues that must be resolved before secure function evaluation and mechanism design can live happily ever after:

- Protocols for secure function evaluation are rather complex and require a lot of interaction between the parties.
- Complexity of mechanism: since the secure function protocol has to emulate the center in some insecure protocol, the computational complexity of the mechanism (in some model) may be a significant factor in the applicability of these results.
- Conditions of security: all constructions are secure as long as a certain coalition of the parties does not collude maliciously. This must match the reality of the application.
- Stability notions: in a cryptographic setting there is a small (hopefully negligible) probability that things will go wrong, e.g. the adversary will guess the secret key. The stability notions such as domination and Nash equilibrium must reflect this relaxation.

In this tutorial we will survey the cryptographic techniques and architectures for implementing protocols for Mechanism Design and examine their applicability and versatility to the problems of Mechanism Design. We will also see what changes have to be made in the equilibrium notions to make them relevant to a cryptographic setting.

2 Cryptographic Tools

We now describes some cryptographic tools that are used in the construction of protocols for secure function evaluation. In particular we use two types of cryptographic tools: pseudo-random functions, and Oblivious Transfer.

2.1 Pseudorandomness

In the cryptographic context pseudorandomness means indistinguishability from random by any *feasible* test. I.e. two ensembles A and B are indistinguishable if no test executed by a polynomial time machine can distinguish with non-negligible advantage whether the input it received came from A or B.

Pseudo-random Functions and Permutations: A pseudo-random function is one that cannot be distinguished from a truly random one by an observer who is given access to the function in a black-box manner. I.e. there is a function F_K specified by a short key K and the observer can only access the function F_K by adaptively specifying inputs and getting the value of the function on these inputs. See [7, 5, 14] for precise definition and various constructions.

Implementation of Pseudo-random Primitives: A common working assumption is that one can model block ciphers (such as AES, or triple DES), or *keyed* one-way hash functions (such as HMAC), as a pseudo-random function. Therefore the function $F_K(x)$ can be implemented by keying a block cipher with the key K and encrypting x, or keying a hash function with K and applying it to x. Under such an assumption the evaluation of a pseudo-random function at a given point is then considerably cheaper than a typical public-key operation such as RSA. The latter typically involve modular exponentiation over large domains.

OT	Sender	Chooser
Input	m_0, m_1	$\sigma \in \{0,1\}$
Output	—	m_σ

Table 1: 1-out-of-2 Oblivious Transfer

2.2 Simulation

One of the important notions investigated in modern cryptography is that of simulation. A protocol is deemed secure if for any possible adversary there is a simulator that outputs similar or indistinguishable distributions to the adversary's ones. The best known example of simulation based security are zero-knowledge proofs [10, 8]. These proofs are important in making sure that parties do not stray from their prescribed protocol.

2.3 Oblivious Transfer

Oblivious Transfer (or OT) is a type of protocol, first suggested by Rabin [21], where one party (the sender) has some input and the other party (the chooser) learns some aspect of the information without 'hinting' which aspect of the information was transferred. More specifically, the following two-party protocol is known as "1-out-of-2 Oblivious Transfer"[1]. The protocol involves two parties, a *Sender* that knows two secret values $\langle m_0, m_1 \rangle$, and a *Chooser* whose input is $\sigma \in \{0,1\}$. At the end of the protocol the Chooser learns m_σ while learning nothing about $m_{1-\sigma}$, and the Sender learns nothing about σ. This is summarized in Table 1. Similarly, it is possible to define 1-out-n OT, where the sender has n secret values.

OT is an essential and sufficient component of secure function evaluation protocols. There are several reasonable implementations of OT protocols. They are all based on public-key operations. However, it turns out that performing 1-out-n OT is not much more expensive than 1-out-2 OT [16, 17].

2.4 Secret Sharing

Suppose that we have some definition of *legal* subsets of users that are allowed certain information. for example, consider the k-out-of-n threshold construction, i.e. all subsets containing at least k users. Secret sharing is a way to split a secret s into n shares such that:

- From any given subset of the shares corresponding to a legal subset of users it is possible to reconstruct the original secret s.
- But if the shares do not contain a legal subset, then no information about s is leaked from the corresponding shares.

For the k-out-of-n threshold a good implementation is via polynomial interpolation.

[1]The notion of 1-out-2 oblivious transfer was suggested by Even, Goldreich and Lempel [4] as a generalization of Rabin's "oblivious transfer" [21].

References

[1] B. Aiello, Y. Ishai and O. Reingold, *Priced Oblivious Transfer: How to Sell Digital Goods*, Advances in Cryptology – Euorocrypt 2001, Springer.

[2] M. Ben-Or, S. Goldwasser and A. Wigderson, *Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation*, Proc. of the ACM Symp. on Theory of Computing, 1988, pp. 1–10.

[3] Y. Dodis, S. Halevi and T. Rabin, *A Cryptographic Solution to a Game Theoretic Problem*, Advances in Cryptology – Crypto '2000, LNCS 1880, Springer, pp. 112–130.

[4] S. Even, O. Goldreich and A. Lempel, *A Randomized Protocol for Signing Contracts*, Communications of the ACM **28**, 1985, pp. 637–647.

[5] O. Goldreich, **Foundations of Cryptography** (Fragments of a Book), 1995. Electronic publication: `http://www.eccc.uni-trier.de/eccc/` (Electronic Colloquium on Computational Complexity). See also `http://www.wisdom.weizmann.ac.il/~oded/foc.html/`

[6] O. Goldreich, *Secure multi-party Computation*, Theory of Cryptography Library, 1998, `http://philby.ucsd.edu/cryptolib/`

[7] O. Goldreich, S. Goldwasser and S. Micali, How to construct random functions, J. of the ACM., vol. 33, 1986, 792–807.

[8] O. Goldreich, S. Micali and A. Wigderson, *Proofs that Yield Nothing But their Validity, and a Methodology of Cryptographic Protocol Design*, J. of the ACM 38, 1991, pp. 691–729.

[9] O. Goldreich, M. Micali and A. Wigderson, *How to play any mental game*, Proc. of the ACM Symp. on Theory of Computing, 1987, pp. 218–229.

[10] S. Goldwasser, S. Micali and C. Rackoff, *The Knowledge Complexity of Interactive Proof-Systems*, Siam J. on Computing, 18(1) (1989), pp 186-208.

[11] M. Hirt and U. Maurer *Complete Characterization of Adversaries Tolerable in Secure Multi-Party Computation*, ACM Symposium on Principles of Distributed Computing, pp. 25–34, 1997.

[12] Y. Ishai and E. Kushilevitz, *Randomizing Polynomials: A new Representation with Applications to Round-Efficient Secure Computation*, Proc. of the IEEE Symp. on Found. of Computer Science, 2000, pp. 294–304.

[13] J. Kilian, *Founding Cryptography on Oblivious Transfer*, Proc. of the ACM Symp. on Theory of Computing, 1988, pp. 20–31.

[14] Luby M., **Pseudorandomness and Cryptographic Applications**, Princeton University Press, 1996.

[15] M. Naor and K. Nissim, *Communication Preserving Protocols for Secure Function Evaluation*, Proc. 33rd ACM Symp. on Theory of Computing, 2001.

[16] M. Naor and B. Pinkas, *Oblivious Transfer and Polynomial Evaluation*, Proc. of the ACM Symp. on Theory of Computing, 1999, pp. 245–254.

[17] M. Naor and B. Pinkas, *Efficient Oblivious Transfer Protocols*, Proc. of 13th ACM-SIAM SODA, pp. 448–457, 2001.

[18] M. Naor, B. Pinkas and R. Sumner, *Privacy preserving auctions and mechanism design*, Proc. of the ACM conference on Electronic Commerce (EC99), pp. 129–139, 1999.

[19] N. Nisan and A. Ronen, *Algorithmic mechanism design*, Proc. 31st ACM Symp. on Theory of Computing, 1999, 129-140.

[20] C. Papadimitriou, *Algorithms, Games and the Internet*, Proc. 33rd ACM Symp. on Theory of Computing, 2001.

[21] M. O. Rabin, *How to exchange secrets by oblivious transfer*, Tech. Memo TR-81, Aiken Computation Laboratory, 1981.

[22] T. Rabin, M. Ben-Or, *Verifiable Secret Sharing and Multiparty Protocols with Honest Majority*, Proc. of the ACM Symp. on Theory of Computing, 1989, pp. 73–85.

[23] H. R. Varian, *Economic mechanism design for computerized agents*, First USENIX Workshop on Electronic Commerce, 1995.

[24] D. Vickrey, *Counter speculation, auctions, and competitive sealed tenders*, Journal of Finance, March 1961, pp. 9–37.

[25] A.C. Yao, *Protocols for Secure Computations*, Proc. of the IEEE Symp. on Found. of Computer Science, 1982, pp. 160–164.

[26] A.C. Yao, *How to generate and exchange secrets*, Proc. of the IEEE Symp. on Found. of Computer Science, 1986, pp. 162–167.