Invited Talk:

# Multi-Party Computations: Past and Present

Shafi Goldwasser*

Cryptography is traditionally concerned with two users, Alice and Bob, who want to communicate privately and authentically over an insecure channel, and in modern times engage in protocols such as coin flipping over the telephone, contract signing, and interactive zero-knowledge proofs.

Today, Alice and Bob are part of the internet. The cryptographic challenges involve $n$ users who want to perform arbitrary interactive and probabilistic computations among themselves, all the while keeping their local data secret.

Examples of such computations, where secrecy is an issue, include:

- *Elections over the internet*: The voters would like to be able to verify that the votes were tallied up correctly without revealing individual votes. In addition, voters should choose votes independently from each other, and the voting authorities should ensure that all votes are by legal voters.

- *Electronic Bidding for Contracts* : Several bidders bid for a work contract over the network, so that the lowest bid verifiably gets the contract. In addition, bids should remain secret, and bidders should choose their individual bids completely independently from each other.

- *Joint Data Base Computation*: Several data bases would like to jointly compute queries which depends on their joint data, without showing each other their entire data base. For example, each data base is a list of private names and the query is which (or how many) names appear on all the lists.

- *Private and Secure Data Base Access*: A client wants to ask queries of a database without the database knowing what has been asked and without the client learning more than the query requested.

- *Joint Signatures*: A group of users would like to be able to digitally sign documents so that only if all of them (or at least a number above threshold) participate, then a signature can be produced.

- *Joint Decryption*: A group of users would like to be able to decrypt messages only if all of them (or at least a number above threshold) participate. This is similar to Split Key Escrow where trustees of the government hold a piece of every users secret decryption key, which would enable them to decrypt users data if and only if they pool their pieces together.

All of the above examples are special cases of the so called *multi-party computation* problem: How to compute *any* probabilistic function on $n$

*MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, MA 02139, USA, and the Weizmann Institue. e-mail: shafi@theory.lcs.mit.edu.

inputs, in a distributed network where each participant holds one of the inputs, ensuring independence of the inputs, correctness of the computation, and that no more information is revealed to participants in the computation than can be computed from a coalition of participants inputs and outputs.

Note that a trivial centeralized solution to this problem would be to assume a trusted center (denoted by TC) exists, and have all users send their inputs to TC to compute their respective outputs. We, however, do not trust any center, and thus desire a distributed solution where trust is distributed.

The history of the multi-party computation problem is extensive since its introduction by Yao [Y] for $n = 2$ and by Goldreich, Micali, and Wigderson [GMW2] for general $n$. I will survey much of this work in the talk, but only mention in brief a few directions of research in this short note. My apologies for the many missing written references.

The first question to consider is: who is the adversary and what are his capabilities? Several adversaries have been considered in the litreature.

- The *Passive* adversary is a coalition of users in the network, who participate in the protocol without deviating from it except for possibly deciding on their inputs together rather than independent of each other before the protocol starts. The goal of the passive adversary is to compute information on inputs of participants who are not part of the coalition. Originally, in [Y,GMW2] the parties were polynomial time bounded. In a line of work initiated by Ben Or, Goldwasser, and Wigderson [BGW] the computational limitations on the adversary were removed, but usere were assumed to be able to communicate in pairs in perfect secrecy.

- The *Byzantine* adversary is a coalition of users in the network who can deviate from the protocol in an arbitrary fashion depending on their inputs and messages received in the protocol. Again, in [Y, GMW2] the adversary is polynomial time bounded, whereas in [BGW, CCD] they are not. The schedualing of faults may be static or dynamic. In static schedualing the faults are fixed in advance, whereas in the dynamic case, users can join the faulty coalition at any time of the protocol depending on messages exchanged so far. Generally, it is much harder to prove results in presence of dynamic schedualing of faults, unless one assumes that non-faulty processors erase their memory at every round [BH]. Proving security for dynamic schedualing without this simplifying assumption was recently achieved by Canetti, Feige, Goldreich, and Naor [CGFN].

- The *Mobile* adversary is a corrupt coalition of users who may be either passive or Byzantine, with the property that at every round of communication of the protocol (for simplicity we only consider here a synchronous network) a different set of users may make up the coalition, with the restriction that at most a fixed number $t$ of users participate in the corrupt coalition at every round. The study of this adversary model was initiated by Ostrovsky and Yung [OY].

- The *Coercing* adversary who can force users to choose their inputs in a way he favors, was introduced in the context of electronic elections by Benaloh and Tunistra [BT], and generalized to arbitrary multi-party computation by Canetti and Genaro [CG].

The next question is how to define precisely what we mean by a secure solution to the multi-party computation problem. Much effort by Micali and Rogaway, Beaver, Goldwasser and Levin, and Canetti [MR, B, GL, C] has been devoted to coming up with crisp definitions of security for multi-party computation which capture the properties which emerge from the above examples. The definition we will use is from [GL] and [C]. Informally, let $TCS_f$ denote the trusted center solution (see above) for a particular probabilistic function $f$. An adversary against a $TCS_f$ is a coalition of users users who can decide on

their inputs before giving them to the trusted center, and similarily can decide on their final outputs after receiving their answers from the trsuted center. We call a multi-party protocol for $f$ *secure (respectively computationaly secure)* for any of the adversary classes above, if for any execution of the protocol and adversary in the class, there exists an adversary against the $TCS_f$ which could achieve the same (respectively polynomial time indistinguishable) output distribution. This definition implies the properties of correctness, input independence and privacy that emerged earlier in the examples.

Many parameters of the network underlying the multi-party computation have been considered. Notably, the "timing" of the network, the number of users in a coalition, availability or lack of broadcast channels. Efficiency parameters considered include the number of rounds of communication necessary to complete the protocol, and the communication complexity of secure computation . Canetti and Rabin [CR] and Ben-Or, Canetti and Goldreich [BCG] achieve multi-party computations in asynchronous networks whereas the original works focused on synchronous networks. The question of whether a broadcast channel is necessary or not was considered by Ben-Or and Rabin [BR] who showed that the number of faults tolerated for general multiparty computation can increase from a $\frac{1}{3}$ minority to a mere minority, if a broadcast hannel was available as a primitive. Franklin and Yung [FR] initiated the study of the communication complexity of uncoditionally secure protocols. The number of rounds of communicatios was studied by Beaver and Bar Ilan [BI] and by Beaver, Micali and Rogaway in the computational setting [BMR].

What type of results have been shown?

The work is generally divided into two categories:

1. Results which make computational assumptions such as the existence of trapdoor functions, do not assume secure channels, and achieve computational security.

2. Results which make physical assumptions such as the existence of perfect secret channels between pairs of users, make no computational assumption, and achieve unconditionally secure privacy. The work of [CGFN] essentially shows a general transformation between results obtained in the perfect secret channel model into results that hold in model without these assumption, by introducing a special encryption functions for sending messages between users over insecure channels.

Remarkably strong completeness theorems are known for the general multi-party computation problem, as indicated in the many papers in the biblipgraphy. Essentially, we know for all the adversary classes outlined above, how given the description of any $n$-input probabilistic function $f$, to automatically construct a secure (resp. computationally secure) multi-party protocol for $f$, as long as the number of total users $n$, and users in a faulty colition obey $n \geq ct + 1$ for optimal values of $c \geq 2$. The value of $c$ changes depending on the adversary class, type of security required (computational or information theoretic), and the parameters of the network, The results in the computational setting require various cryptographic assumptions such as the existence of one-way functions and oblivious transfer protocols. The case of a coalition of half or more of faulty users has been studied ([Y,C, BG,BL]), but it seems much harder to solve in a completely satisfactory manner. The underlying problem in this case is how to release the outputs of the computation via a slow probabilistic process, to prevent a majority of faulty users to quit the protocol early as soon as they compute their own outputs. If we assume that none of the users stop early, Kilian shows under the assumption that a primitive for oblivious transfer exists, how to achieve secure two party protocols, and [GL] show under the same assumption how to achieve security for $n$ users with a faulty majority.

What are the tools and techniques underlying the solutions?

The beauty of multi-party protocols is that they use a rich body of tools and sub-protocols, some of which developed especially for this application and some previously developed for the

cryptographic non-distributed setting. They include Zero-Knowledge Proofs [GMR, GMW1], Probabilistic Encryption [GM], error correcting code representation of data [BGW], various distributed commitment schemes, and Oblivious Transfer [MRa1, K, CK]. Most importantly, *secret sharing* and *computing with shares of a secret* is fundamental to achieving secure multiparty computation. In particular, the polynomial secret sharing of Shamir [Sh] in the case of passive adversary is a corner stone in multiparty computations, and the verifiable secret sharing of Chor, Goldwasser, Micali and Awerbuch [CGMA, RB] plays an analogous role in the Byzantine adversary case. Beautiful techniques have been developed to *compute* on shares of secrets for the various secret sharing scheme, and in essence such a technique is present in any work on multi-party computation, e.g. [GMW2, CCD, BGW, BCC].

Whereas in the 80's the focus of research was to show the most general result possible yielding multi-party protocol solutions fo any probabilistic function, any adversary class, and any network constraints, the theme of the 90's is different. Much of current work is to focus on *efficient* and *non-interactive* solutions to special important problems such as joint-signatures, joint-decryption, and secure and private data base access. Some of the new conceptual issues that researchers are currently tackling are the deniability of users actions in presence of a coercing adversary and the anonymity of users.

We believe that the field of multi party computations is today where public-key cryptography was ten years ago, namely an extremely powerful tool and rich theory whose real-life usage is at this time only beginning but will become in the future an integral part of our computing reality.

## References


[Be] D. Beaver, "Foundations of Secure Interactive Computing", *CRYPTO,* 1991.

[BH] D. Beaver and S. Haber, "Cryptographic Protocols Provably secure Against Dynamic Adversaries", *Eurocrypt,* 1992.

[BR1] M. Bellare and P. Rogaway, "Entity authentication and key distribution", *Advances in Cryptology: Proc. of Crypto 93,* August 1993, pp. 232-249.

[BT] Josh Benaloh and Dwight Tunistra, "Receipt-Free Secret-Ballot Elections", *26th STOC,* 1994, pp. 544-552.

[BCG] M. Ben-Or, R. Canetti and O. Goldreich, "Asynchronous Secure Computations", *25th STOC,* 1993, pp. 52-61.

[BGW] M. Ben-Or, S. Goldwasser and A. Wigderson, " Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation", *20th STOC,* 1988, pp. 1-10.

[BKR] M. Ben-Or, B. Kelmer and T. Rabin, "Asynchronous Secure Computation with Optimal Resilience", *13th PODC,* 1994 pp. 183-192.

[BL] M. Ben-Or and N. Linial, "Collective Coin Flipping", *Advances in Computing Research, Vol. 5,* 1989, pp. 91-115.

[BCC] G. Brassard, D. Chaum and C. Crepeau, "Minimum Disclosure Proofs of Knowledge", *Journal of Computing and System Sciences,* Vol. 37, No. 2, 1988, pp. 156-189.

[C] R. Canetti, "Asynchronous Secure Computation", *Technical Report no. 755,* CS department, Technion, 1992.

[CK] C. Crepeau and J. Kilian, "Achieving Oblivious Transfer Using Weakened security Assumptions", *29th FOCS,* pp. 42-52.

[CDNO] R. Canetti, C. Dwork, M. Naor and R. Ostrovsky, "Deniable Encryptions", manuscript.

[CFGN] R. Canetti, U. Feige, O. Goldreich and M. Naor, "Adaptively Secure Computation", *28th STOC,* 1996.

[CG] R. Canetti and R. Genaro, "Deniable Multiparty Computation", manuscript.

[CaH] R. Canetti and A. Herzberg, "Maintaining security in the presence of transient faults", *Crypto' 94*, 1994, pp. 425-439.

[CR] R. Canetti and T. Rabin, "Optimal Asynchronous Byzantine Agreement", *25th STOC*, 1993, pp. 42-51.

[CCD] D. Chaum, C. Crepeau and I Damgard, "Multiparty unconditionally secure protocols", *20th STOC*, 1988, pp. 11-19.

[CGMA] B. Chor, S. Goldwasser, S. Micali and B. Awerbuch, "Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults", *26th FOCS*, 1985, pp. 383-395.

[CK] B. Chor and E. Kushilevitz, "A Zero-One Law for Boolean Privacy", *SIAM J. on Disc. Math., Vol. 4, no. 1*, 1991, pp.36-47.

[CM] B. Chor and L. Moscovici, "Solvability in Asynchronous Environments", *30th FOCS*, 1989.

[DH] W. Diffie and M. Hellman, "New directions in cryptography", *IEEE Trans. on Info. Theory*, IT-22(6), 1976, pp. 644-654.

[DDN] D. Dolev, C. Dwork and M. Naor, "Nonmalleable Cryptography", *23rd STOC*, 1991.

[FM] P. Feldman and S. Micali, "An Optimal Algorithm For Synchronous Byzantine Agreement", *20th STOC*, 1988, pp. 148-161.

[FY] M. Franklin and M.Yung, "Communication Complexity of Secure Computation", *24th STOC*.

[G] O. Goldreich, "Foundations of Cryptography (Fragments of a Book)", ed. Dept. of Computer Science and Applied Mathemetics, Weizmann Institute, 1995.

[GGL] O. Goldreich, S. Goldwasser, and N. Linial, "Fault-Tolerant Computation in the Full Information Model", *32nd FOCS*, 1991, pp. 447-457.

[GMW1] O. Goldreich, S. Micali and A. Wigderson, "Proof that Yield Nothing But their Validity and a Methodology of Cryptographic Protocol Design", *27th FOCS*, 1986, pp. 174-187.

[GMW2] O. Goldreich, S. Micali and A. Wigderson, "How to Play any Mental Game", *19th STOC*, 1987,pp. 218-229.

[GM] S. Goldwasser, and S. Micali, " Probabilistic Encryption", *SOTC*, 1982. Journal version appeared in the JCSS special issue.

[GwL] S. Goldwasser, and L. Levin, "Fair Computation of General Functions in Presence of Immoral Majority", *CRYPTO*, 1990.

[GMR] S. Goldwasser , S. Micali and C. Rackoff, "The Knowledge Complexity of Interactive Proof Systems", *SIAM Journal on Comput.*, Vol. 18, No. 1, 1989, pp. 186-208.

[HJKY] A. Herzberg, S. Jarecki, H. Krawczyk and M. Yung, "Proactive Secret Sharing or: How to Cope with Perpetual Leakage", *CRYPTO* 1995.

[K] J. Kilian, "Founding Cryptography on Oblivious Transfer", *STOC 88*,, pp. 20-31.

[MR] S. Micali and P. Rogaway, "Secure Computation". Preliminary version in *CRYPTO 91.*

[OY] R. Ostrovsky and M. Yung, "How to withstand mobile virus attacks", *Proceedings of the* $10^{th}$ *Annual ACM Symposium on Principles of Distributed Computing*, 1991, pp 51–59.

[MRa1] M. Rabin, "How to exchange secrets by oblivious transfer", Tech. Memo TR-81, Aiken Computation Laboratory, Harvard U., 1981.

[RB] T. Rabin and M. Ben-Or, "Verifiable Secret Sharing and Multiparty Protocols with Honest Majority", *21st STOC,* 1989, pp. 73-85.

[TRa] T. Rabin , "Robust Sharing of Secrets When The Dealer is Honest or Faulty", *Journal of the ACM*, No. 6, Vol. 41, 1994, pp. 1089-1109.

[SK] K. Sako and J. Kilian, "Receipt-Free Mix-Type Voting Scheme", *Eurocrypt* 1995, pp. 393-403.

[Sh] A. Shamir, "How to share a secret", *CACM, Vol. 22, No. 11,* 1979, pp. 612-613.

[Y1] A. Yao, "Protocols for Secure Computation", *23th FOCS,* 1982, pp. 160-164.