Brief Announcement: On Secure m-Party Computation, Commuting Permutation Systems and Unassisted Non-Interactive MPC

Navneet Agarwal
Indian Institute of Technology Bombay
navneet@cse.iitb.ac.in

Sanat Anand
Indian Institute of Technology Bombay
sanat@cse.iitb.ac.in

Manoj Prabhakaran
Indian Institute of Technology Bombay
mp@cse.iitb.ac.in

A fundamental problem in the theory of secure multi-party computation (MPC) is to characterize functions with *more than 2 parties* which admit MPC protocols with information-theoretic security against passive corruption. This question has seen little progress since the work of Chor and Ishai [2], which demonstrated difficulties in resolving it.

We report an ongoing work, in which we make significant progress towards resolving this question in the important case of *aggregating functionalities*: In an aggregating functionality, there are m parties $P_1, \ldots, P_m$ with inputs $x_1, \ldots, x_m$ and an aggregating party P_0 must learn $f(x_1, \ldots, x_m)$. Aggregating functionalities form a practically and theoretically important class. In particular, it has been the subject of an influential line of study that started with the *minimal model for secure computation* of Feige, Kilian and Naor [4]. This model – sometimes refered to as the Private Simultaneous Messages (PSM) model – served as a precursor of important concepts like randomized encodings [5] that have proven useful in a variety of cryptographic applications. Recently, a strengthening of this model, called Non-Interactive MPC (NIMPC) was introduced by Beimel et al. [1], which is closer to standard MPC in terms of the security requirements. In both these models the severe restriction on the communication pattern often leads to simple and elegant protocols. Indeed, for specialized functions (like "Remote-OT" and AND) the original protocols developed in the PSM model [4] can also be shown to be optimal (or very nearly so) in terms of communication and randomness complexity [3, 9]. Similarly, Beimel et al. discovered several elegant NIMPC protocols for special classes of functions [1]. However, these protocols do not directly translate to MPC protocols *as these models include a trusted party* which sends correlated random variables to the parties in a pre-processing phase. The term aggregating functionality was coined in [8].

Our contributions in this work fall into three broad categories: (1) minimal models of computation, (2) algebraic-combinatorial classes of aggregating functionalities, and (3) positive and negative results relating the above two.

Minimal Models of MPC. The previous minimalistic models of MPC – PSM [4] and NIMPC [1] – admit secure protocols for all functions, unlike the full-fledged MPC model. Our minimalistic models (called UNIMPC* and UNIMPC) admit secure protocols only for functions which have secure protocols in the MPC model. While the previous models were proposed in the context of studying communication complexity of information-theoretic MPC, ours is perhaps the first significant model aimed at studying the feasibility of information-theoretic MPC.

UNIMPC stands for *Unassisted NIMPC* and, as the name suggests, removes the assist-

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Daniel Marx, and Don Sannella; Article No. 501; pp. 501:1–501:3
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

ance from the trusted party in NIMPC: Instead the parties should securely compute the correlated randomness by themselves, in an offline phase. Unlike PSM and NIMPC, which allow trusted parties, *UNIMPC retains the standard security model of MPC*, allowing corruption of any set of parties. While MPC and NIMPC are incomparable in the sense that an MPC protocol does not yield an NIMPC protocol (because of the general communication pattern) and an NIMPC protocol does not yield an MPC protocol (because of the use of a trusted party), UNIMPC could be seen as a common denominator of these two secure computation models.

> *A UNIMPC protocol can be directly interpreted as an MPC protocol as well as an NIMPC protocol.*

UNIMPC* corresponds to a minimalistic version of UNIMPC, with protocols which have a single round of (simultaneous) communication among the parties before they get their inputs, followed by a single message from each party to the aggregator after they receive their input. (UNIMPC allows arbitrarily many rounds of communication prior to receiving inputs.) Understanding the gap between the classes of functionalities with UNIMPC and UNIMPC* protocols is closely related to understanding the power of multiparty secure sampling [7].

Commuting Permutations Systems. We identify an algebraic-combinatorial structure called Commuting Permutations System (CPS) and a sub-class called Commuting Permutation Subgroup Systems (CPSS).

Below S_n denotes the symmetric group – the group of all permutations of n elements.

▶ **Definition 1.** An (n, m)-*Commuting Permutations System* (CPS) is a collection $(X_1, \cdots, X_m)$ where for all $i \in [m]$, $X_i \subseteq S_n$ contains the identity permutation, and for any collection $(\pi_1, \cdots .\pi_m)$ with $\pi_i \in X_i$, and $\rho \in S_m$, $\pi_1 \circ \cdots \circ \pi_m(1) = \pi_{\rho(1)} \circ \cdots \circ \pi_{\rho(m)}(1)$.[1]

It is called an (n, m)-*Commuting Permutation Subgroups System* (CPSS) if each X_i is a subgroup of S_n.

An $(m + 1)$-party aggregating functionality $f : X_1 \times \cdots \times X_m \to [n]$ is said to be a *CPS functionality* (resp. CPSS functionality) if $(X_1, \cdots, X_m)$ is an (n, m)-CPS (resp. (n, m)-CPSS) and for all $(\pi_1, \cdots, \pi_m) \in X_1 \times \cdots \times X_m$, $f(\pi_1, \cdots, \pi_m) = (\prod_{i\in[m]} \pi_i)(1)$.

Results. Our main results can be summarized as follows. Writing CPS (or CPSS) for class of functionalities that "embed" into a CPS (respectively, CPSS) functionality, and UNIMPC*, UNIMPC and MPC for classes of functionalities that admit the corresponding secure protocol, we have, for any number of parties,

$$\text{CPSS} \Rightarrow \text{UNIMPC}^* \Rightarrow \text{UNIMPC} \Rightarrow \text{MPC} \Rightarrow \text{CPS}.$$

Note that we leave an intriguing gap between the necessary and sufficient conditions. In particular we leave open the possibility that the set of functionalities with UNIMPC protocols is a strict subset of the set of aggregating functionalities with MPC protocols, and is a strict superset of aggregating functionalities with UNIMPC* protocols. However, these differences disappear for small number of parties: When the number of input parties is 2, we show that UNIMPC* ⇔ CPS, and when the number of input parties is 3, UNIMPC ⇔ CPS.

---

[1] Choice of 1 is arbitrary. Requiring identity permutation to always be part of each X_i is w.l.o.g., as a CPS without it will remain a CPS on adding it.

We also obtain a characterization of all "Latin hypercube functionalities" which have an MPC protocol, and show that they all have UNIMPC* protocol. This result relies on the above results, as well as on the existence of NIMPC protocols for every CPS functionality. For the sake of being self-contained we present a simple NIMPC protocol for general functionalities, which in fact turns out to be more efficient than the prior constructions [1, 6].

Our results could be seen as a step towards fully characterizing the functionalities with information-theoretic MPC protocols in various security models. For instance, for characterizing functionalities with UC secure protocols, aggregating functionalities remain the only class to be understood [8], and the sub-classes of aggregating functionalities identified in this work can serve as a starting point for understanding UC security. Similarly, the problem of characterizing symmetric functions (when all parties get the same output) as considered in [2] is still unsolved, but our positive results do present new possibilities there (because a passive-secure MPC protocol for an aggregating functionality can be readily converted into one for a symmetric functionality computing the same function).

References

1. Amos Beimel, Ariel Gabizon, Yuval Ishai, Eyal Kushilevitz, Sigurd Meldgaard, and Anat Paskin-Cherniavsky. Non-interactive secure multiparty computation. In Juan A. Garay and Rosario Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*, volume 8617 of *Lecture Notes in Computer Science*, pages 387–404. Springer, 2014. URL: `https://doi.org/10.1007/978-3-662-44381-1_22`, `doi:10.1007/978-3-662-44381-1_22`.
2. Benny Chor and Yuval Ishai. On privacy and partition arguments. *Information and Computation*, 167(1):2–9, 2001.
3. Deepesh Data, Manoj Prabhakaran, and Vinod Prabhakaran. On the communication complexity of secure computation. CoRR Report 1311.7584 available from `http://arxiv.org`, 2013.
4. Uriel Feige, Joe Kilian, and Moni Naor. A minimal model for secure computation (extended abstract). In *STOC*, pages 554–563. ACM, 1994.
5. Yuval Ishai and Eyal Kushilevitz. Randomizing polynomials: A new representation with applications to round-efficient secure computation. In *FOCS*, pages 294–304, 2000.
6. Satoshi Obana and Maki Yoshida. An efficient construction of non-interactive secure multiparty computation. In *Cryptology and Network Security - 15th International Conference, CANS 2016, Milan, Italy, November 14-16, 2016, Proceedings*, pages 604–614, 2016.
7. Manoj Prabhakaran and Vinod Prabhakaran. On secure multiparty sampling for more than two parties. In *Proceedings of the 2012 IEEE International Information Theory Workshop (ITW 2012)*, 2012.
8. Manoj Prabhakaran and Mike Rosulek. Cryptographic complexity of multi-party computation problems: Classifications and separations. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 262–279. Springer, 2008.
9. Sundara Rajan S, S. Rajakrishnan, A. Thangaraj, and V. Prabhakaran. Lower bounds and optimal protocols for three-party secure computation. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 1361–1365, July 2016.