Hidden Δ-fairness: A Novel Notion for Fair Secure Two-Party Computation

Saskia Bayreuther Robin Berger Felix Dörre
Jeremias Mechler Jörn Müller-Quade

Karlsruhe Institute of Technology

36th Crypto Day, 14/15 March 2024

Secure two-party computation (2PC) allows two mutually distrusting parties to compute a joint function over their inputs, guaranteeing properties such as *input privacy* or *correctness*.

For many tasks, such as joint computation of statistics, it is important that when one party receives the result of the computation, the other party also receives the result. Unfortunately, this property, which is called *fairness*, is unattainable in the two-party setting for arbitrary functions Cleve (1986). In some settings, *e.g.* when computing only certain functions, complete fairness is achievable Gordon *et al.* (2008); Choudhuri *et al.* (2017); Cohen *et al.* (2022). Weaker variants have been proposed such as *partial fairness* Gordon & Katz (2010); Bailey *et al.* (2022), *gradual release fairness* Blum (1983), and *fairness with penalties* Bentov & Kumaresan (2014). All these notions are free from any measure of time and in case of a premature abort of the adversary, do not necessarily guarantee output delivery to the honest party.

Another fairness notion, that includes the concept of time to 2PC, proposed by Pass *et al.* (2017) is called Δ-*fairness*. Informally, it guarantees that, even if the adversary aborts prematurely and receives the output in round r, the honest party receives the output by round $\Delta(r)$. This notion is achieved by using so-called *secure enclaves* through the Generalized Universal Composition (GUC) framework $\mathcal{G}_{att}$. In comparison to cryptographic tools like garbeled circuits, which are commonly used in 2PC, with secure enclaves, most of the complexity vanishes.

In many settings, Δ-fairness is not sufficient, because a corrupt party is *guaranteed* to receive its output before the honest party, giving the corrupt party an advantage in further interaction. Worse, as Δ is known to the corrupt party, it can abort the protocol when it is most advantageous.

We extend the concept of Δ-fairness by introducing a new fairness notion, which we call *hidden* Δ-*fairness*, which addresses these problems. First, under our new notion, a corrupt party may not benefit from aborting, because it only learns the result first with a probability of 1/2. Moreover, Δ and other parameters are sampled according to a given distribution and remain unknown to the participants in the computation.

We propose a 2 PC protocol that achieves hidden Δ-fairness, also using secure enclaves via $\mathcal{G}_{att}$, and prove its security in the GUC framework.

crypto day matters 36 (2024)
Beiträge vom 36. Kryptotag, 14.-15. MÃ¤rz 2024, Regensburg.
doi:10.18420/cdm-2024-36-32

References

Bolton Bailey *et al.* (2022). General partially fair multi-party computation with VDFs. IACR ePrint.

Iddo Bentov & Ranjit Kumaresan (2014). How to use bitcoin to design fair protocols. In *CRYPTO*.

Manuel Blum (1983). How to exchange (secret) keys. *TOCS* .

Arka Rai Choudhuri *et al.* (2017). Fairness in an unfair world: Fair multiparty computation from public bulletin boards. In *CCS*.

Richard Cleve (1986). Limits on the security of coin flips when half the processors are faulty. In *STOC*.

Ran Cohen *et al.* (2022). From fairness to full security in multiparty computation. *Journal of Cryptology* .

S Dov Gordon & Jonathan Katz (2010). Partial fairness in secure two-party computation. In *EUROCRYPT*.

S. Dov Gordon *et al.* (2008). Complete fairness in secure two-party computation. IACR ePrint.

Rafael Pass *et al.* (2017). Formal abstractions for attested execution secure processors. In *EUROCRYPT*.