

Complete Fairness in Cryptographic Protocols

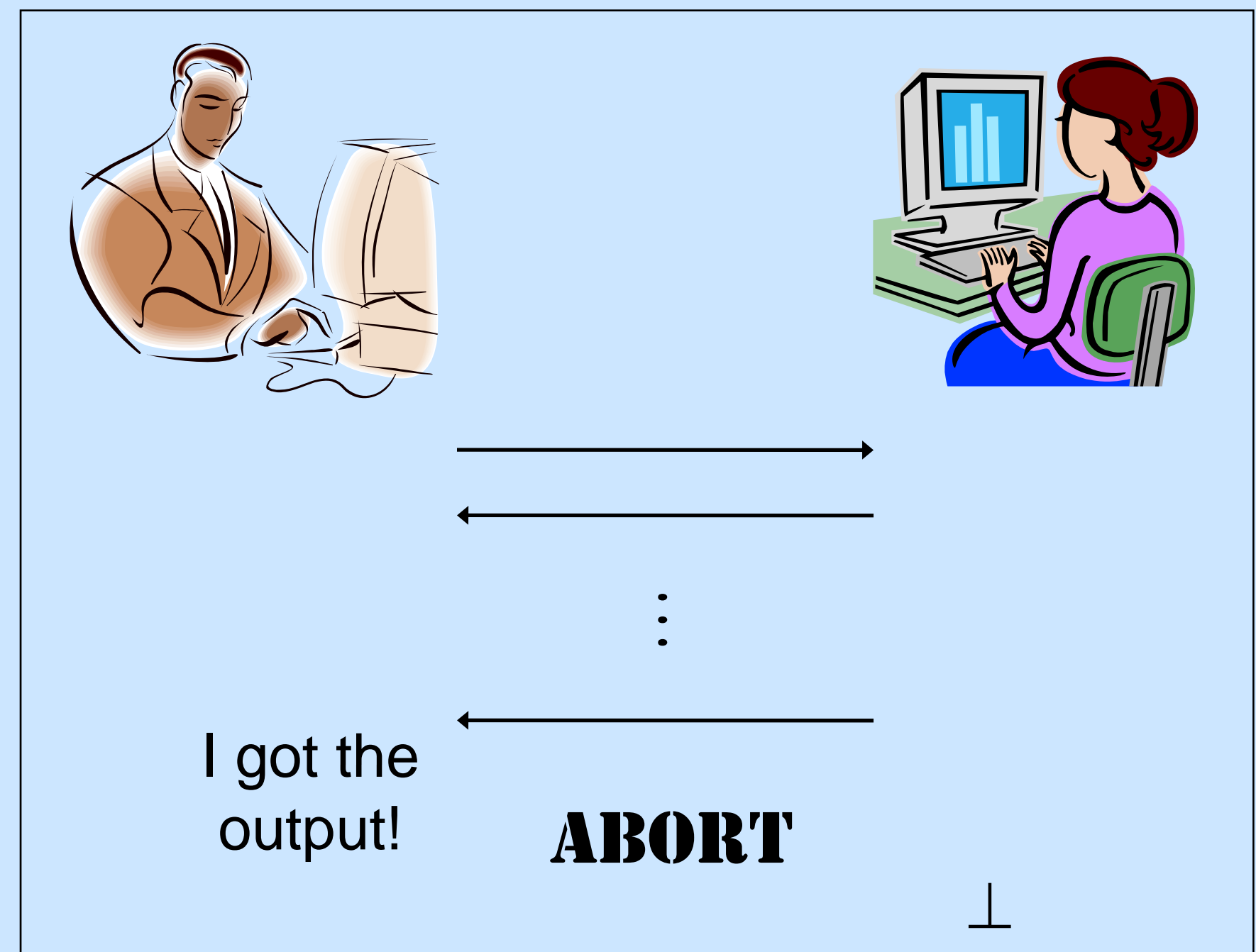
Jonathan Katz, University of Maryland



Fairness

In the setting of secure multi-party computation, a group of mutually-distrusting parties P_1, \dots, P_n , holding input x_1, \dots, x_n , respectively, wish to compute $y = f(x_1, \dots, x_n)$ while maintaining desirable properties such as privacy, correctness, etc. A protocol is *fair* if either *everyone* learns the output y , or else *no one* does.

Fairness is important in many specific contexts (e.g., contract signing, secure negotiation), and implies that an adversary cannot mount a “denial of service attack” that prevents honest parties from receiving the output



Fairness is known to be impossible *in general* when there is no honest majority. For over 20 years, the prevailing belief was that fairness is *always* impossible in this case (i.e., for any “interesting” function f). Is this folklore belief true?

Is complete fairness possible?

Intuitively, one party must learn the output first; what if this party aborts the protocol immediately thereafter?

Approach and Impact

New approach

- We ask whether there are *any* non-trivial functions for which complete fairness is possible
- Alternately, can we prove impossibility?

Research Impact

- We show the *first* completely-fair protocols for non-trivial functions (in the two-party and multi-party settings)
- Forces a re-evaluation of our understanding of fairness!

We introduce two new approaches for constructing fair two-party protocols.

- In our first approach, the round in which a party learns the output *depends on its input*. If a party P_i aborts the protocol in some round, the other party “assumes” that P_i just learned the output, and takes P_i 's input to be the corresponding value. We prove that for certain functions (those without an *embedded XOR*), this approach works.
- In our second approach, the views of each party are correlated random variables that converge geometrically to the correct output. Intuitively, the only way fairness can be violated is if a party aborts in the first round where he learns the correct output; however, neither party can tell for sure when this has occurred. This approach can be shown to work even for certain functions having an embedded XOR.