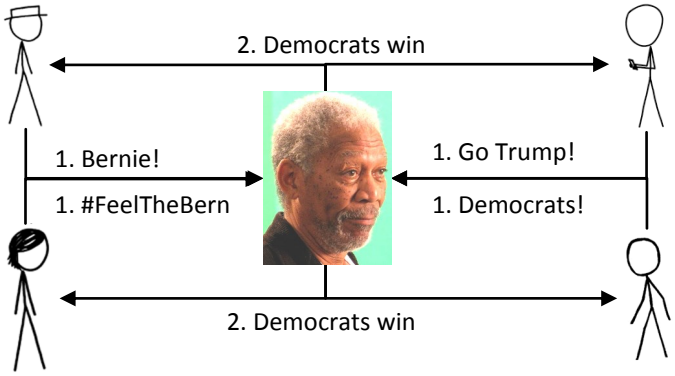
Efficient Multiparty Computation with Identifiable Abort

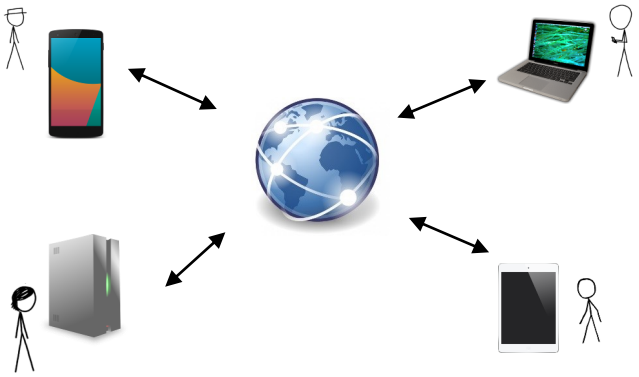
Carsten Baum, Emmanuela Orsini & Peter Scholl

What is Multiparty Computation? An Example: Voting

Dream Election

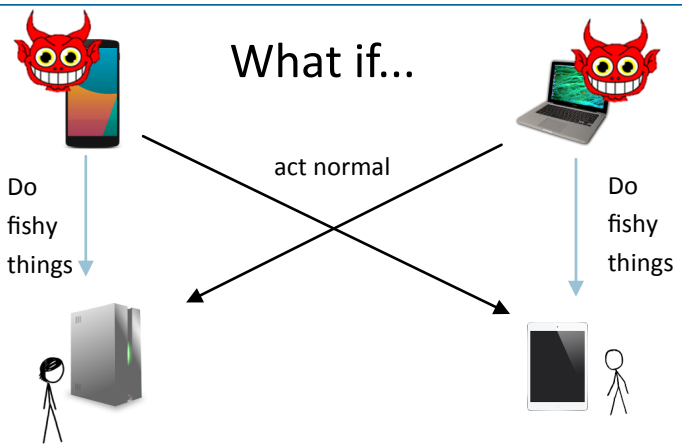


Real World



Goal: Compute a function on inputs from different parties by sending messages over a network. Do this so that the inputs remain private, even if someone arbitrarily deviates from the protocol.

What if...



- both and disagree on who is a bad guy
- even worse: cannot just trust what says because she may also be a corrupt party!
- the problem is also called MPC with Identifiable Abort: If something goes wrong, then we want to identify a cheater

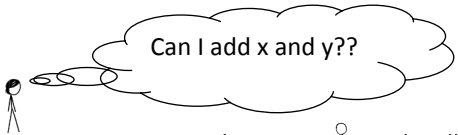
Current solutions to the problem:

[1] gives a theoretical solution where every party proves in every step that it behaved honestly (inefficient)

[2] just assumes to abort in this case (this opens up the protocol to denial of service attacks)

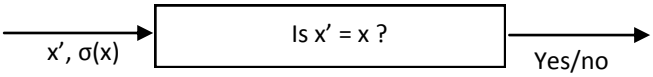
Our Key Idea

- for each value x that has, knows a value σ(x) and has a τ(x). Neither nor knows x though
- when computes f(x) from x, then and can compute σ(f(x)), τ(f(x)) respectively



Yes! Because can locally obtain a σ(x+y) from σ(x), σ(y)! can also compute τ(x+y)

has a magic box: When he gets a value x' from he can check that:



Everyone has this box and all get the same answer! We show how to do this very fast and solve the above problem.

Bibliography:

[1] Yuval Ishai,, Rafail Ostrovsky & Vassilis Zikas: "*Secure Multiparty Computation with Identifiable Abort*", CRYPTO 2014

[2] Ivan Damgård, Valerio Pastro, Nigel Smart and Sarah Zakarias: "*Multiparty computation from somewhat homomorphic encryption*", CRYPTO 2012

[3] Carsten Baum, Emmanuela Orsini and Peter Scholl: "*Efficient Secure Multiparty Computation with Identifiable Abort*", to appear