FairSwap: How to fairly exchange digital goods

- Full Version: https://eprint.iacr.org/2018/740

Motivation

- fair exchange: a sender sells a digital commodity x for a price p to a receiver
- The receiver is willing to pay price p for x if $\phi(x) = 1$, where ϕ is a predicate function.
  - e.g., $x\phi(x) = 1$ if hashing x results into some fixed hash value ($H(x) = h$)
- How can we make sure that seller receives the payment when he delivers x to receiver s.t. $\phi(x) = 1$?
- How can we guarantee that receiver only needs to pay the money if x is indeed correct?

Previous Works

- using trusted third party
- using smart contract
  - it has an important drawback if x is large.
  - for storing x of size 1MB in ETH, the parties would need to pay more than USD 500 in transaction fees.
- using zero-knowledge contingent payments (ZKCP)
  - ZKCP puts significant computational burden on the receiver and the sender

Contribution

1. An efficieint fair exchange of digital goods using smart contracts
   - design simple smart contracts that can be executed with low fees
     - works for arbitrary predicate functions ϕ and large size x
   - avoiding expensive cryptographic tools such as zero-knowledge proofs
2. Proof of Misbehavior
   - originally proposed in the Delegation of Computation ([CRR11] @ CCS11)
   - for large x, proving that seller behaves correctly is very costly
   - however, it is *much cheaper* to instead prove that seller behaved incorrectly.
3. Applications
   - selling files over the Internet
   - Claim-or-refund