Scaling Bitcoin to Support Privacy-Preserving Smart Contracts

Ranjit Kumaresan (MIT)

ranjit@csail.mit.edu

people.csail.mit.edu/ranjit/

Goal of this Talk

- Smart contracts – Scaling
  - Expressivity & Limitations
  - Efficiency
  - Privacy
  - Remove limitations via a natural relaxation
- Highlight: Off-chain crypto for scaling
  - Magic tech: ***Secure Computation***
    - Active research pushing this to practice
  - Integration with Bitcoin backed by academic research
    - Presents new perspectives on scaling issues
  - Encourage more research/engineering/hacking

Smart Contracts

- Contracts
  - Well-defined set of rules among group of agents
  - Rules agreed upon if deemed fair by all agents
  - E.g.: Nuptial agreements, Tax treaties, *Bitcoin*
- Enforcing contracts
  - Typically by some authority (e.g., legal)
  - Typically involves data and/or money

- ***Smart contracts*** via decentralized digital currencies
  - Eliminates authority (and associated costs)
  - Automatic enforcement via consensus

Smart Contracts - Expressivity

- Via scripts
- Support multi-sigs, etc.
- Restrict some OP_CODES

- Via scripts
- Turing-complete!

Later: Both possibly face fundamental limitations

Smart Contracts - Efficiency

- Script verification fast because of restrictions
- Block size restriction does not support scaling wrt number of agents or wrt complexity of contract

- Turing-complete scripts too powerful
- Miners may lose the incentive to verify transactions containing complex scripts

Later: More efficiency metrics for smart contracts

Smart Contracts - Privacy

- Emphasis on consensus
- No native support

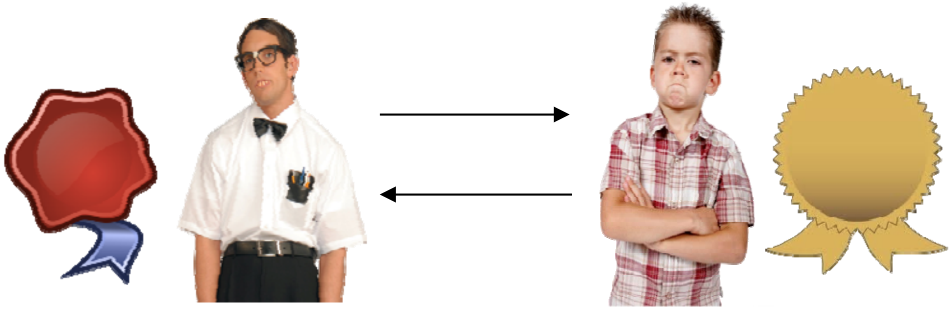
- No native support
- No privacy logic

Later: Off-chain crypto for privacy & more!

Smart Contracts - Limitations

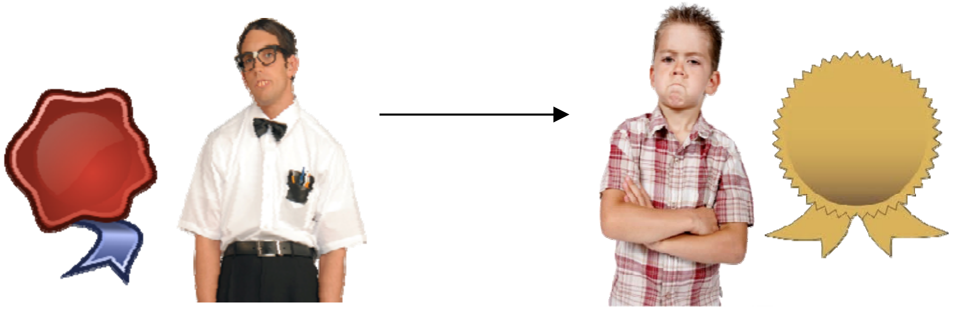
FAIR EXCHANGE

Parties want to exchange digital assets



Abort Attacks

Need to force exchange to happen simultaneously



Fair *currency* exchange

- Use *TierNolan protocol*
- Generally, easy if asset has supporting blockchain

Arbitrary assets

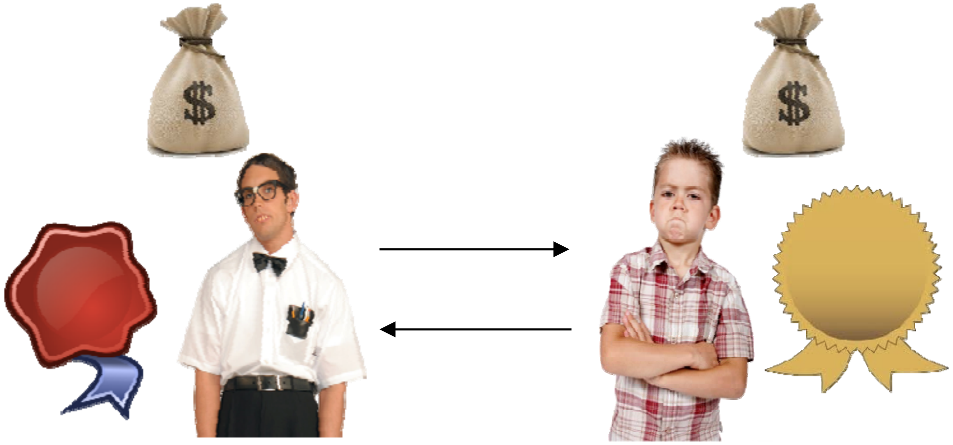
- Don’t know!
- Impossible?

Smart Contracts - Relaxations

FAIR EXCHANGE WITH PENALTIES

Parties want to exchange digital assets;

Upon abort, penalty imposed on cheater



Possible?

- Yes! Even for arbitrary assets [Bentov-Kumaresan'14]
- Protocol uses scripts supported in Bitcoin
  - Scaling issue: Scales poorly in the multi-party setting

Smart Contracts with Penalties

- Add extra ***penalty*** rule in contract
  - Cheating agent pays a penalty to all other agents
- Natural relaxation for contracts
  - Contracts implicitly associated with penalty for “breaking the contract” (e.g.: penalty decided in a court of law)
  - Here: Explicit penalty by associating monetary value
- Allows overcoming fundamental limitations
  - Backed by academic research [ADMM14,BK14,KB14,KBM15]

Example App: Decentralized Poker

- The ***POKER*** “smart contract with penalties”
  - Agents = Players
  - Rules = Poker rules
  - Action steps:
    - Data = Cards
    - Transactions = Bets

- Player may abort in the middle if it’s unlikely to win
  - If player aborts during its action step, then it pays penalty to all other players

Scaling Issues

- Scaling parameters:
  - Number of agents
  - Size of rules
  - Size of data
  - Privacy
    - Contract data typically sensitive
    - Not a good idea to add contract data to the blockchain

Block size limit has direct relevance

- Solution ideas:
  - Try to build complex contracts from ***simpler contracts***
  - Use ***off-chain crypto technology*** to support scaling

Simple Contracts: Claim-or-refund

- Claim-or-refund
  - *Zero-knowledge Contingent Payment* (BTC wiki 2011)
  - 2-party contracts between sender and receiver
  - Sender locks coins in the transaction and specifies criteria
  - Receiver can claim coins within time t by producing data D that satisfies criteria
  - If unclaimed by time t, coins refunded to sender

- Blockchain independent abstraction

- Can build complex contracts from claim-or-refund!!
  - Example: Multiparty Fair Exchange with Penalties

Multi-Party Fair Exchange with Penalties

$$P_1 \xrightarrow[q,\tau]{T} P_2$$

denotes

P_2 must reveal data T within time τ to claim coins(q) from P_1

$$\begin{array}{ccc}
P_1 & \xrightarrow[q,\tau_n]{T_1 \wedge \cdots \wedge T_n} & P_n \\
P_2 & \xrightarrow[q,\tau_n]{T_1 \wedge \cdots \wedge T_n} & P_n \\
 & \vdots & \\
P_{n-2} & \xrightarrow[q,\tau_n]{T_1 \wedge \cdots \wedge T_n} & P_n \\
P_{n-1} & \xrightarrow[q,\tau_n]{T_1 \wedge \cdots \wedge T_n} & P_n
\end{array}$$

$$\begin{array}{ccc}
P_n & \xrightarrow[(n-1)q,\tau_{n-1}]{T_1 \wedge \cdots \wedge T_{n-1}} & P_{n-1} \\
P_{n-1} & \xrightarrow[(n-2)q,\tau_{n-2}]{T_1 \wedge \cdots \wedge T_{n-2}} & P_{n-2} \\
 & \vdots & \\
P_3 & \xrightarrow[2q,\tau_2]{T_1 \wedge T_2} & P_2 \\
P_2 & \xrightarrow[q,\tau_1]{T_1} &
\end{array}$$

Issues

- No data privacy!
- Transactions are 2-party but size grows with n; size also depends on data

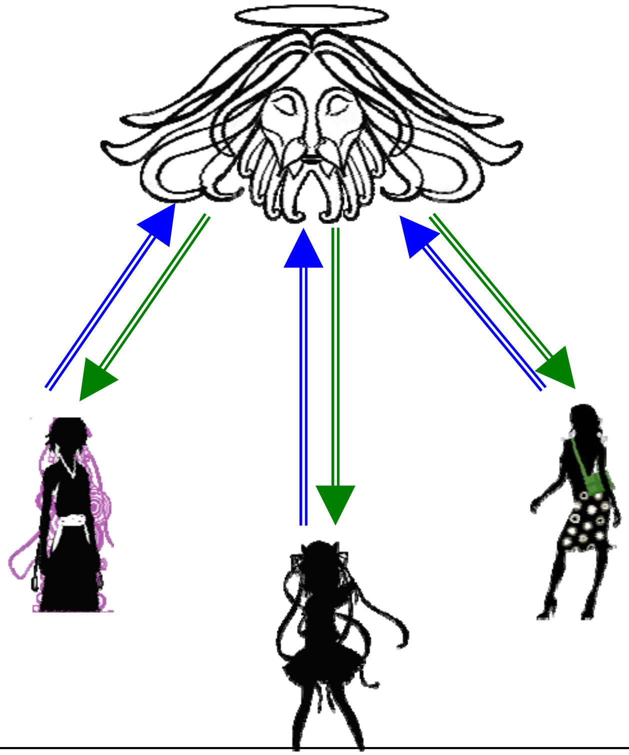
Magic Technology: Secure Computation

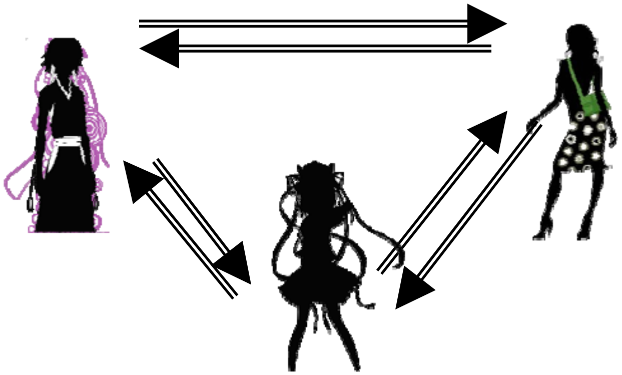
IDEAL

- Parties submit data
- Parties get back results

IDEAL → REAL

- No trusted party!
- Run secure computation protocol!
  - **GOD → CRYPTO**
- Same effect as the IDEAL protocol
  - Privacy/Correctness
- Active area of research
  - Moving from theory to practice!





SNARK, NIZK, FHE, Obfuscation, etc., are special cases of secure computation and impose restrictions on interaction (and are less efficient)

Powerful Combination: Claim-or-refund + Secure Computation

Scaling parameter	Stateless Contracts (Example: Fair exchange)
Number of agents	Decoupled from block size restriction
Size of rules	No on-chain dependence
Size of data	No on-chain dependence
Privacy	Yes

- Get nontrivial feasibility result for *stateful* smart contracts
  - Privacy Preserving
- Caveat: Assumes extended script support for Bitcoin
  - Example: For ***POKER*** smart contract with penalties
    - Need verification of signatures on arbitrary (but bounded data)…. Don’t need Turing-complete scripts
- Another caveat: large number of ordered transactions
  - Use off-chain payment channel like *Lightning*

Academic Work on Bitcoin + Sec.Comp.

- **A Note on Coin Tossing**
  - Back-Bentov (arXiv 2014)
- **Secure Multiparty Computations on Bitcoin**
  - Andrychowicz *et al.* (IEEE S&P 2014 – *best paper*)
- **How to Use Bitcoin to Design Fair Protocols**
  - Bentov-Kumaresan (IACR Crypto 2014)
- **How to Use Bitcoin to Incentivize Correct Computations**
  - Kumaresan-Bentov (ACM CCS 2014)
- **How to Use Bitcoin to Play Decentralized Poker**
  - Kumaresan-Moran-Bentov (ACM CCS 2015)
- **Hawk: The Blockchain Model of Cryptography & Privacy Preserving Smart Contracts**
  - Kosba *et al.* (ePrint 2015)

Summary

- Smart contracts with penalties
  - Removes limitations on expressivity
- Highlight: Off-chain crypto for scaling
  - Magic tech: ***Secure Computation***
    - Active research pushing this to practice
  - Integration with Bitcoin backed by academic research
    - New perspectives on scaling: *Extended script support*
  - Need more research/engineering/hacking

Thank You!

ranjit@csail.mit.edu

people.csail.mit.edu/ranjit/