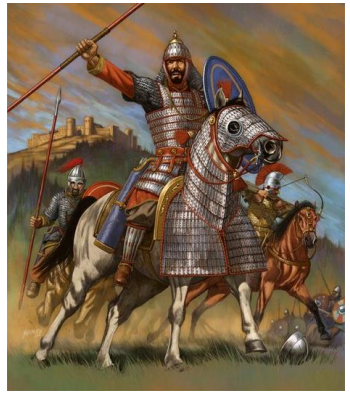
BFTCBFTP: BYZANTINE-FAULT-TOLERANT CONSTRUCTION OF BFT PROTOCOLS

EDWARD TREMEL

SIGSEGV 2019

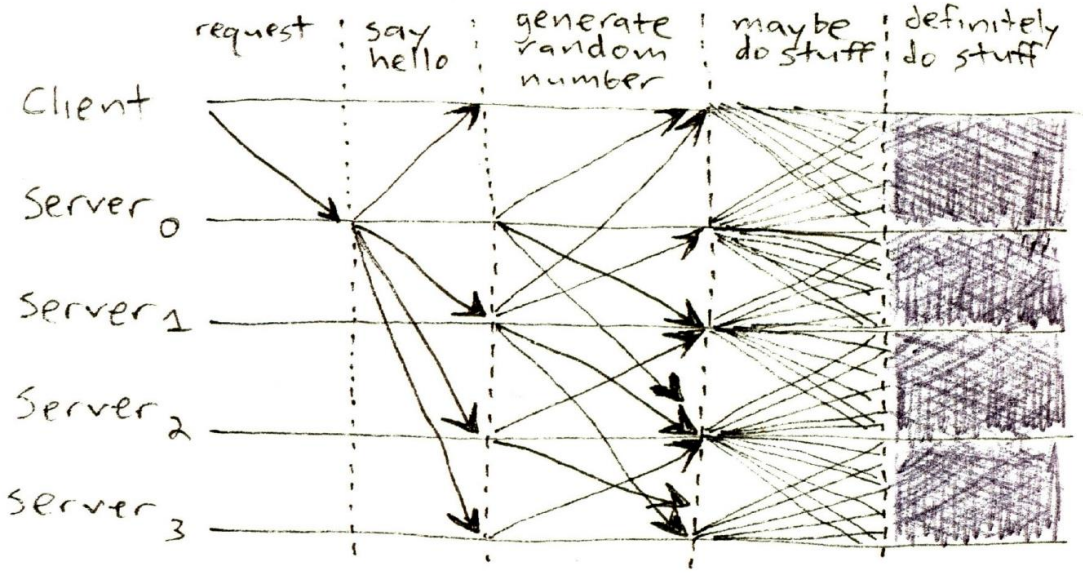
BYZANTINE FAULT TOLERANCE

- Long-standing problem in systems
- Byzantine (adj): excessively complicated, and typically involving a great deal of administrative detail
- Inspired by bickering generals
- Assumes everyone is untrustworthy



BFT PROTOCOLS

- Need to be very complicated
- Much disagreement on how to construct them
- Necessary in order to make fault-tolerant systems
  - Because we said so



PROBLEM: CONSTRUCTION OF NEW BFT PROTOCOLS

- Clearly, we always need more BFT protocols
- Constructing a BFT protocol takes a lot of work, hard for one researcher to do alone
- Distributing the work to multiple researchers would help, but systems researchers bicker more than Byzantine generals

Thema: Byzantine-Fault-Tolerant Middleware for Web-Service Applications*

Michael G. Merideth†, Arun Iyengar‡, Thomas Mikalsen‡, Stefan Tai‡, Isabelle Rouvellou‡, Priya Narasimhan†

† School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
{mgm,priya}@cs.cmu.edu

‡ IBM TJ Watson Research Center, Hawthorne, NY, USA
{aruni,tommi,stai,rouvellou}@us.ibm.com

Fast Byzantine Consensus

Jean-Philippe Martin and Lorenzo Alvisi, *Senior Member, IEEE*

DepSpace: A Byzantine Fault-Tolerant Coordination Service

Alysson Neves Bessani†, Eduardo Pelison Alchieri†, Miguel Correia†, Joni da Silva Fraga‡

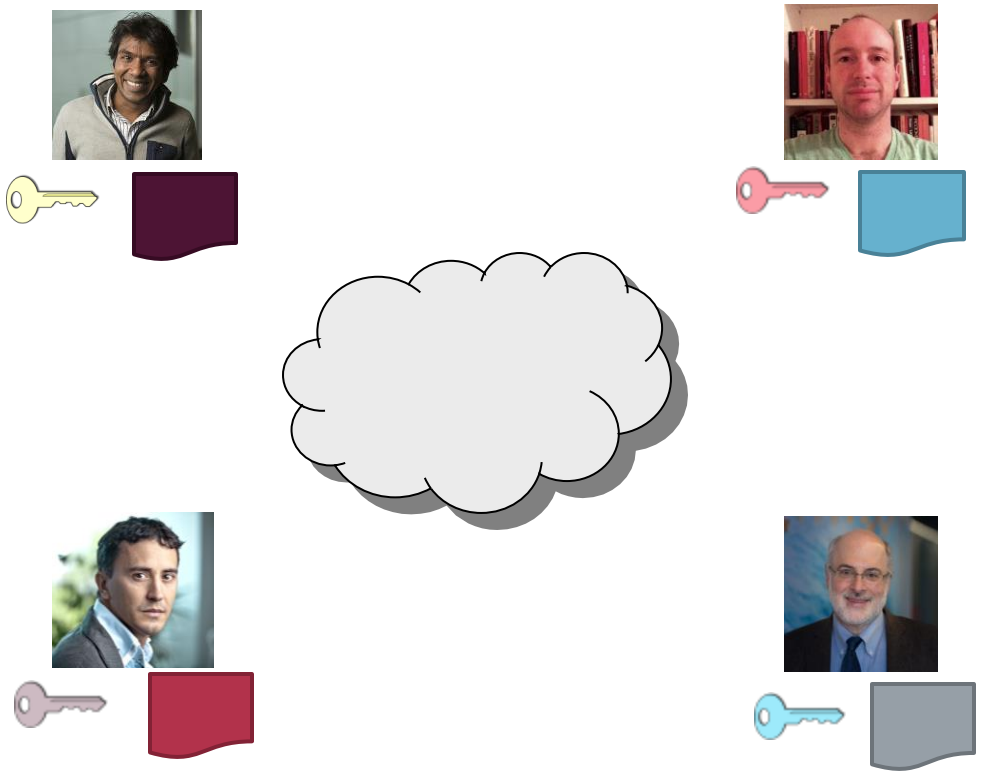
†LaSIGE, University of Lisbon, Lisbon, Portugal
‡DAS, Federal University of Santa Catarina, Florianópolis, Brazil

Zyzzyva: Speculative Byzantine Fault Tolerance

Ramakrishna Kotla, Lorenzo Alvisi, Mike Dahlin, Allen Clement, and Edmund Wong
Dept. of Computer Sciences
University of Texas at Austin
{kotla,lorenzo,dahlin,aclement,elwong}@cs.utexas.edu

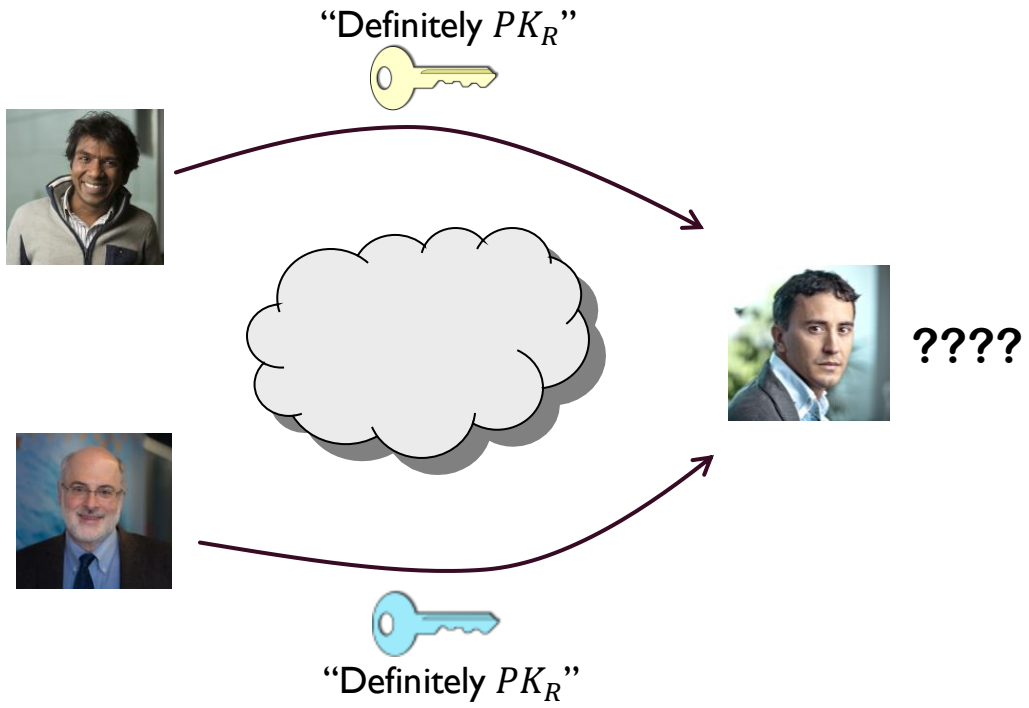
SETUP

- 3f + 1 systems researchers
  - Why? Because that's the standard for BFT
- Mutually distrustful
- Must agree on details of protocol
- No "trusted third party"
- Solution can't have a leader – everyone wants to be the leader
- Everyone has their own public/private key



PREREQUISITE: KEY EXCHANGE

- All BFT protocols depend on signing messages
- Bootstrapping problem: exchanging public keys when the network is untrusted
- Our solution: Researchers meet IRL at a systems conference, give each other keys
- Body doubles impersonating researchers is out-of-scope for this work

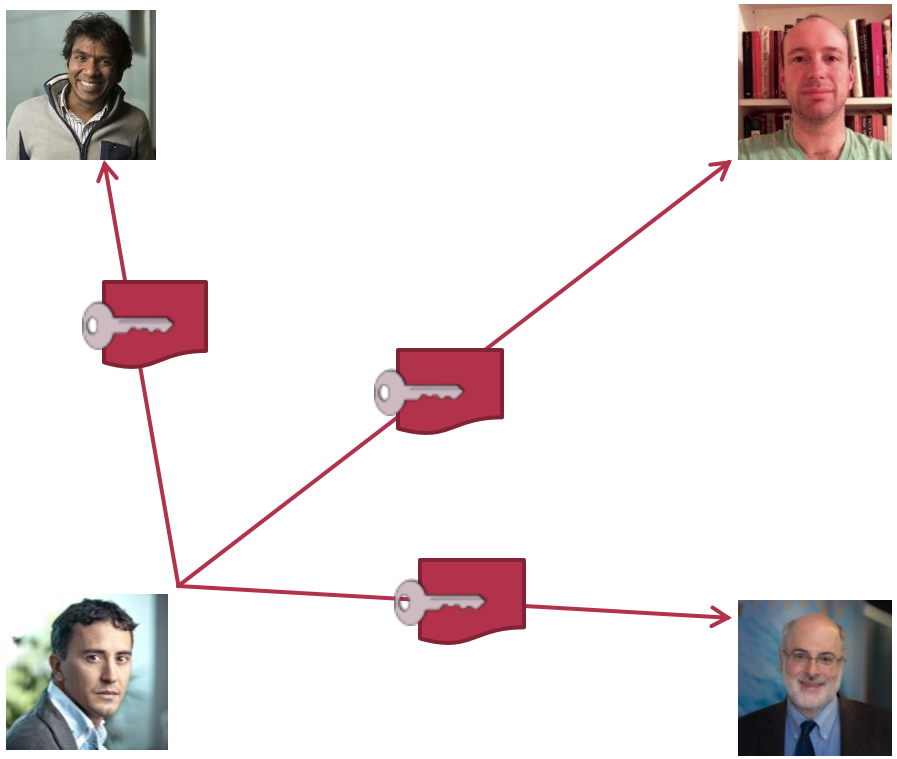


PREREQUISITE: KEY EXCHANGE

- All BFT protocols depend on signing messages
- Bootstrapping problem: exchanging public keys when the network is untrusted
- Our solution: Researchers meet IRL at a systems conference, give each other keys
- Body doubles impersonating researchers is out-of-scope for this work

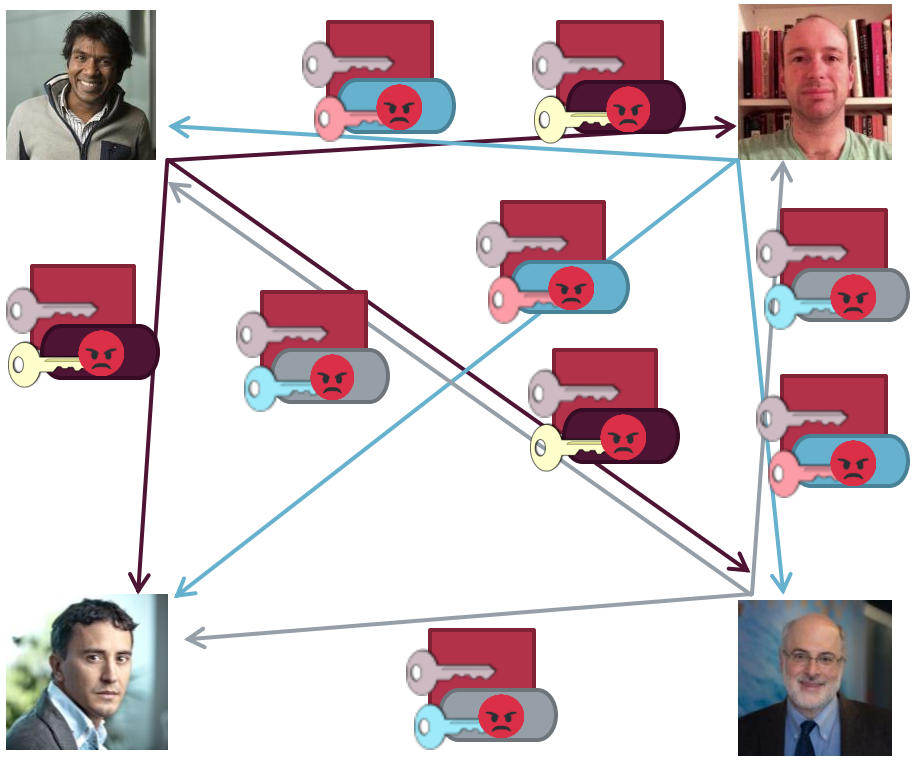
STEP 1: BROADCAST STEP 1 OF PROTOCOL

- Someone broadcasts their proposal for Step 1 of protocol, signed with their private key
- Whoever takes initiative gets to start



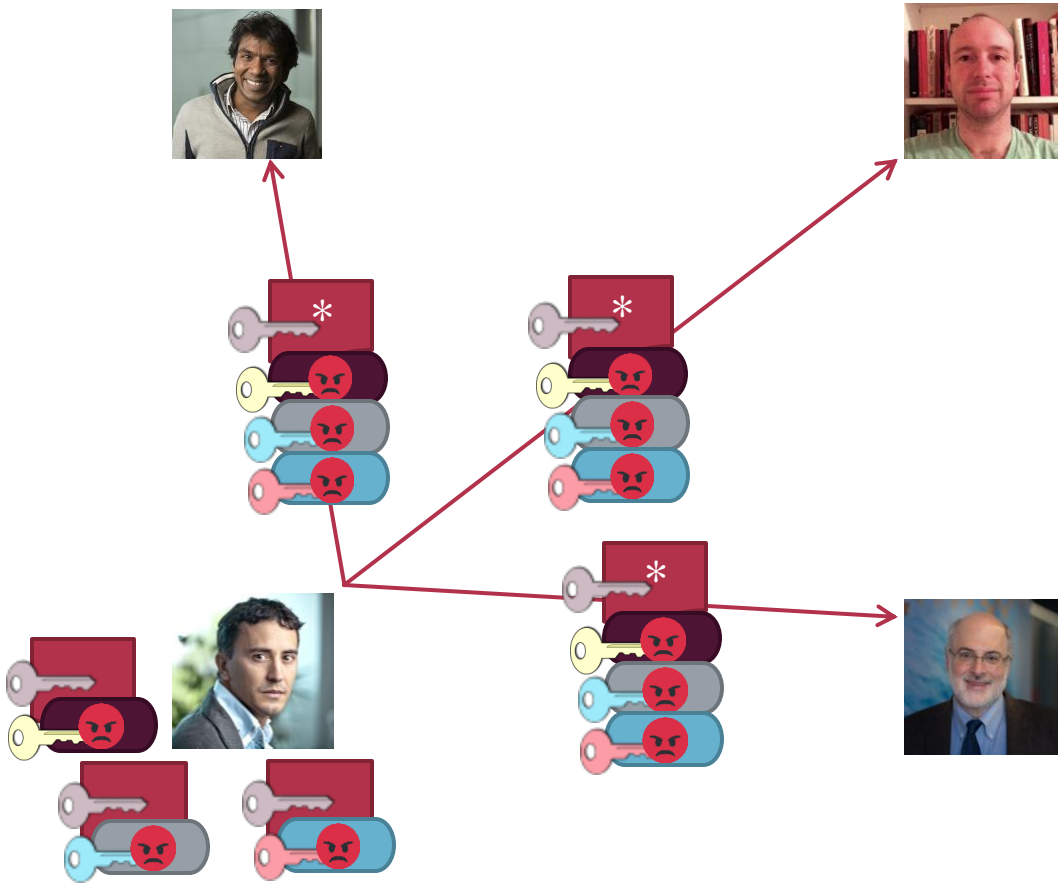
STEP 2: CRITICIZE STEP 1 OF PROTOCOL

- Each other researcher reads Step 1, writes criticism
- Append signed criticism to Step 1, broadcast to other researchers
- If anyone receives criticism with different Step 1, proof that author of Step 1 equivocated



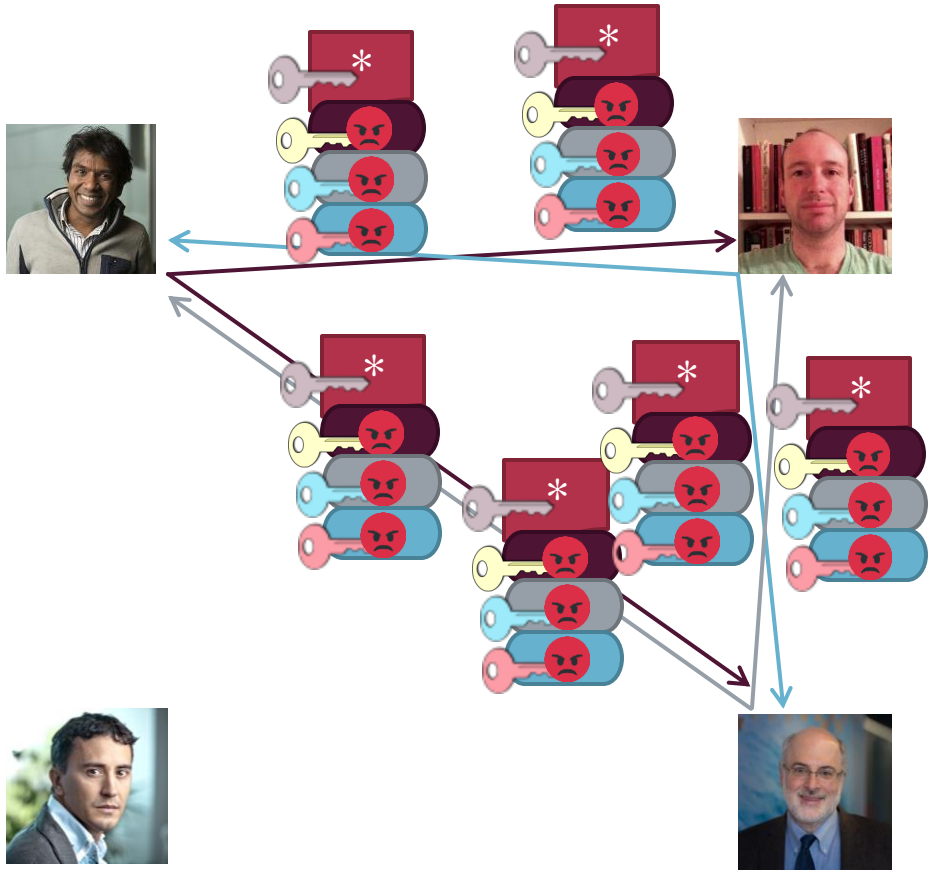
STEP 3: HANDLE CRITICISM

- Author of Step 1, upon accumulating signed criticism from others, may revise step 1 in response
- If criticism is contradictory, may choose to *reject*
- If any criticism agrees, may *begrudgingly accept* and apply to protocol
- Sends out *revised* Step 1, with signed criticism appended, to prove authenticity of criticism
- Critics may detect equivocation by other researchers on their criticism at this point



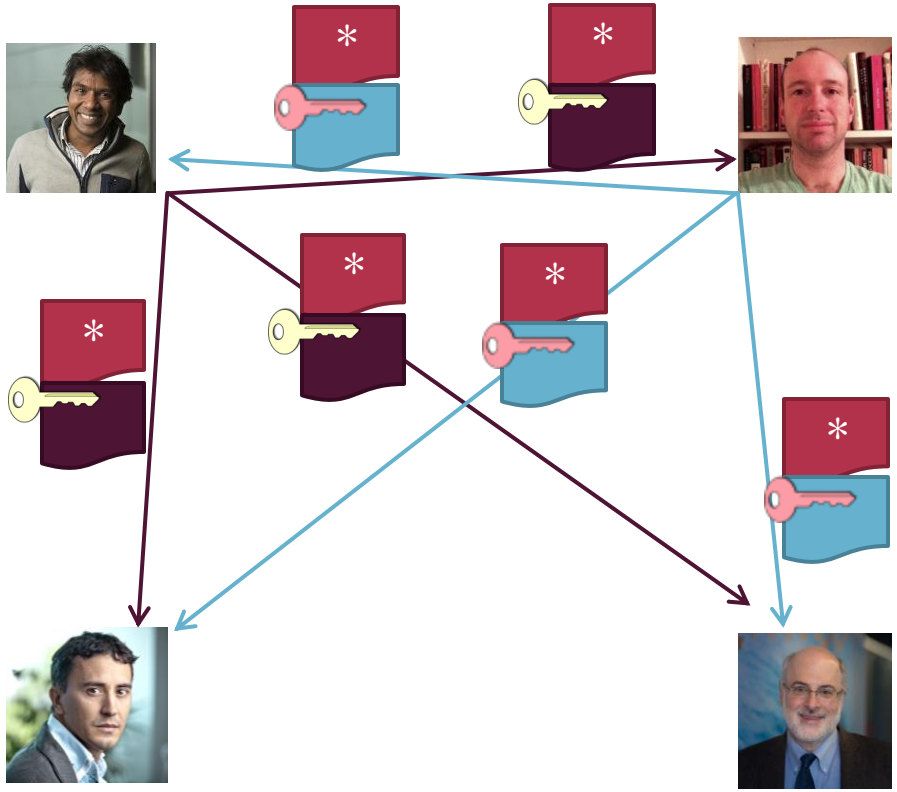
STEP 4: REBROADCAST REVISED STEP 1

- Other researchers echo the revised Step 1 to each other, to ensure author is not equivocating



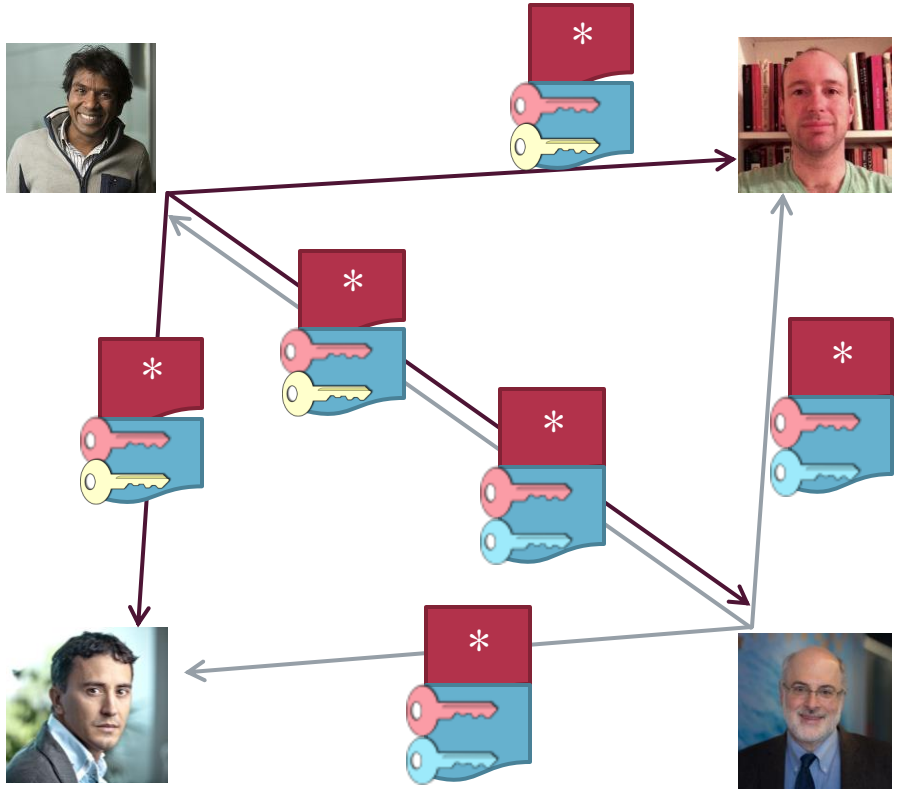
STEP 5: BROADCAST STEP 2 OF PROTOCOL

- Everyone who has an idea for Step 2 broadcasts it, appended to revised Step 1, signed with their key
- Now we have to agree on whose idea to use



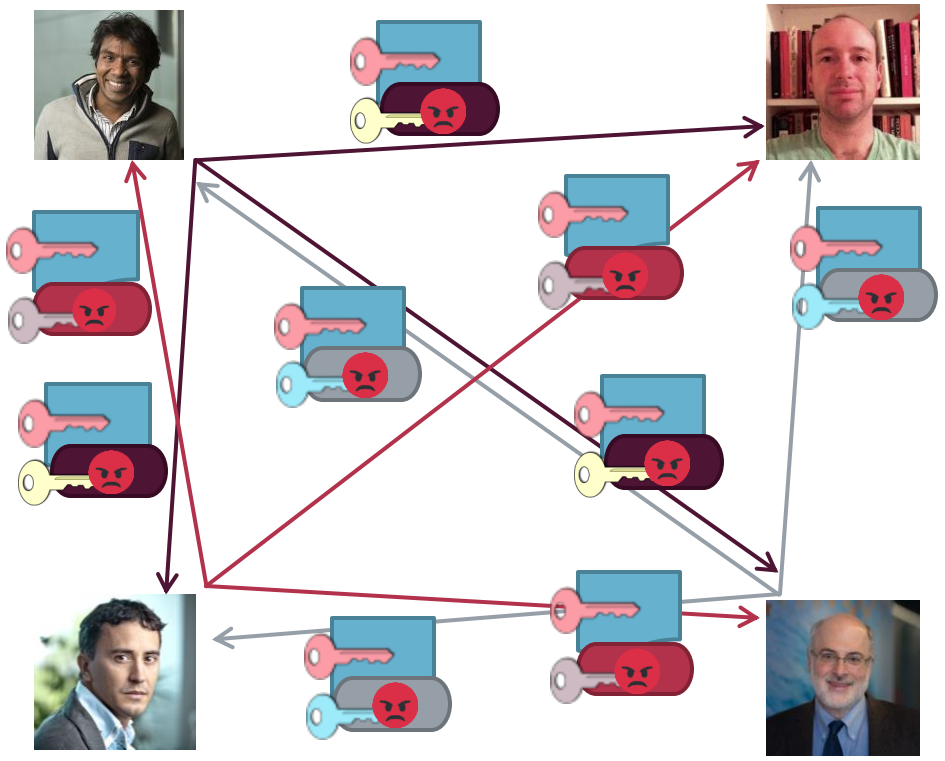
STEP 6: VOTE ON STEP 2

- Researchers sign and rebroadcast a version of Step 2 if they agree to use it
- Once a version of Step 2 has signatures from a majority, continue with it
- Decide whether to vote for a version based on *reputation system*
  - Vote for proposal if you like the researcher who proposed it

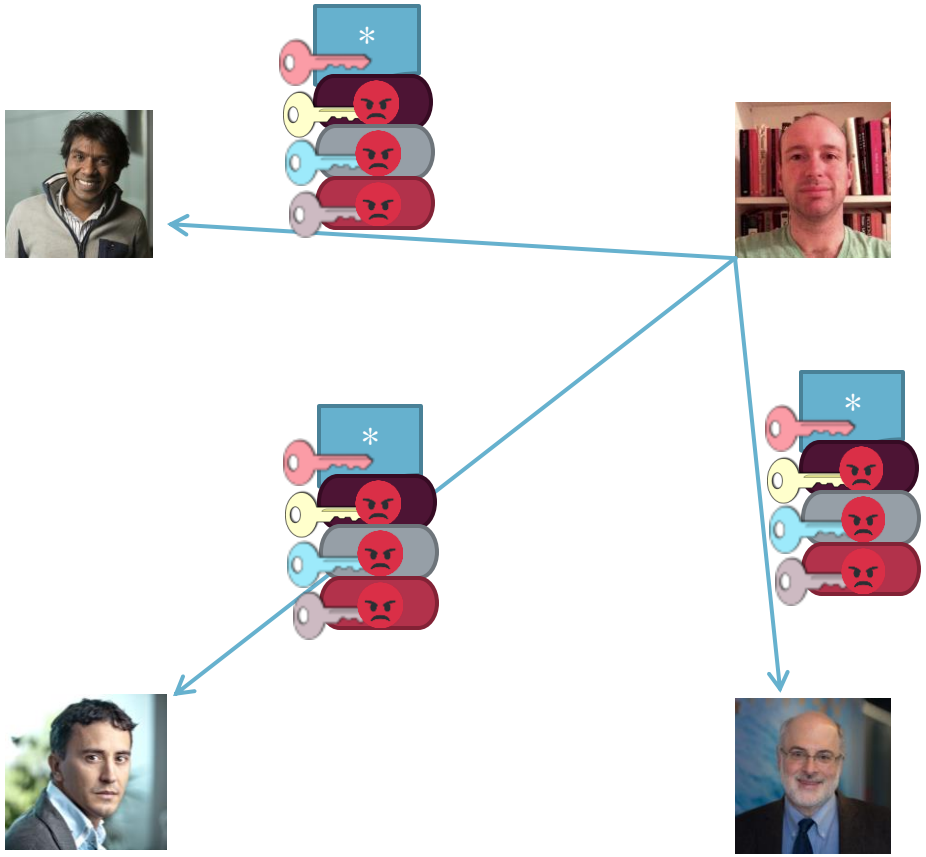


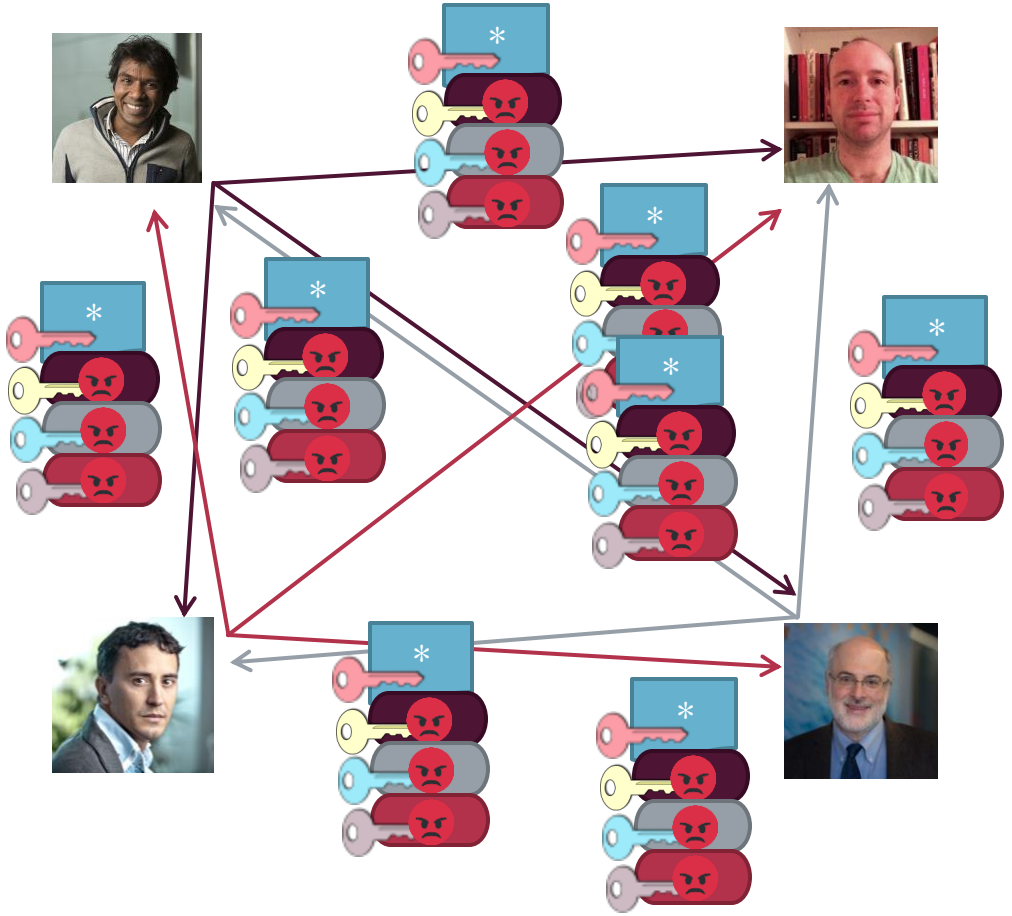
STEP 7: CRITICIZE STEP 2 OF PROTOCOL

- Just like criticism on Step 1
- Criticize version of Step 2 with majority votes

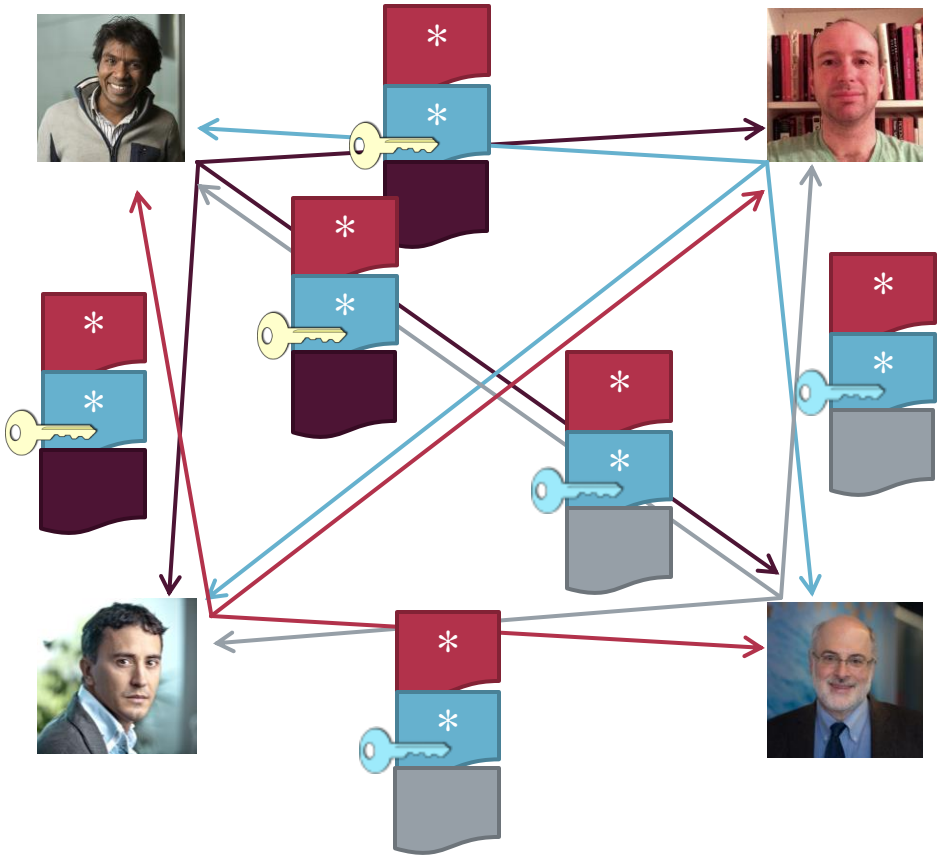


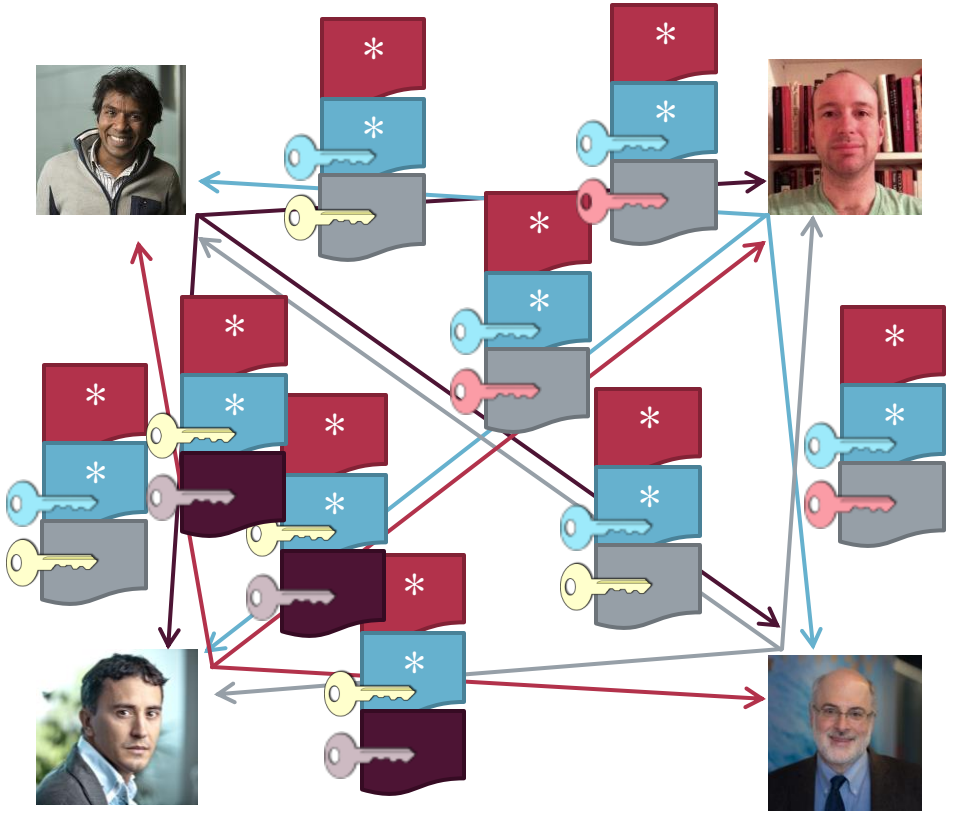
…AND SO ON





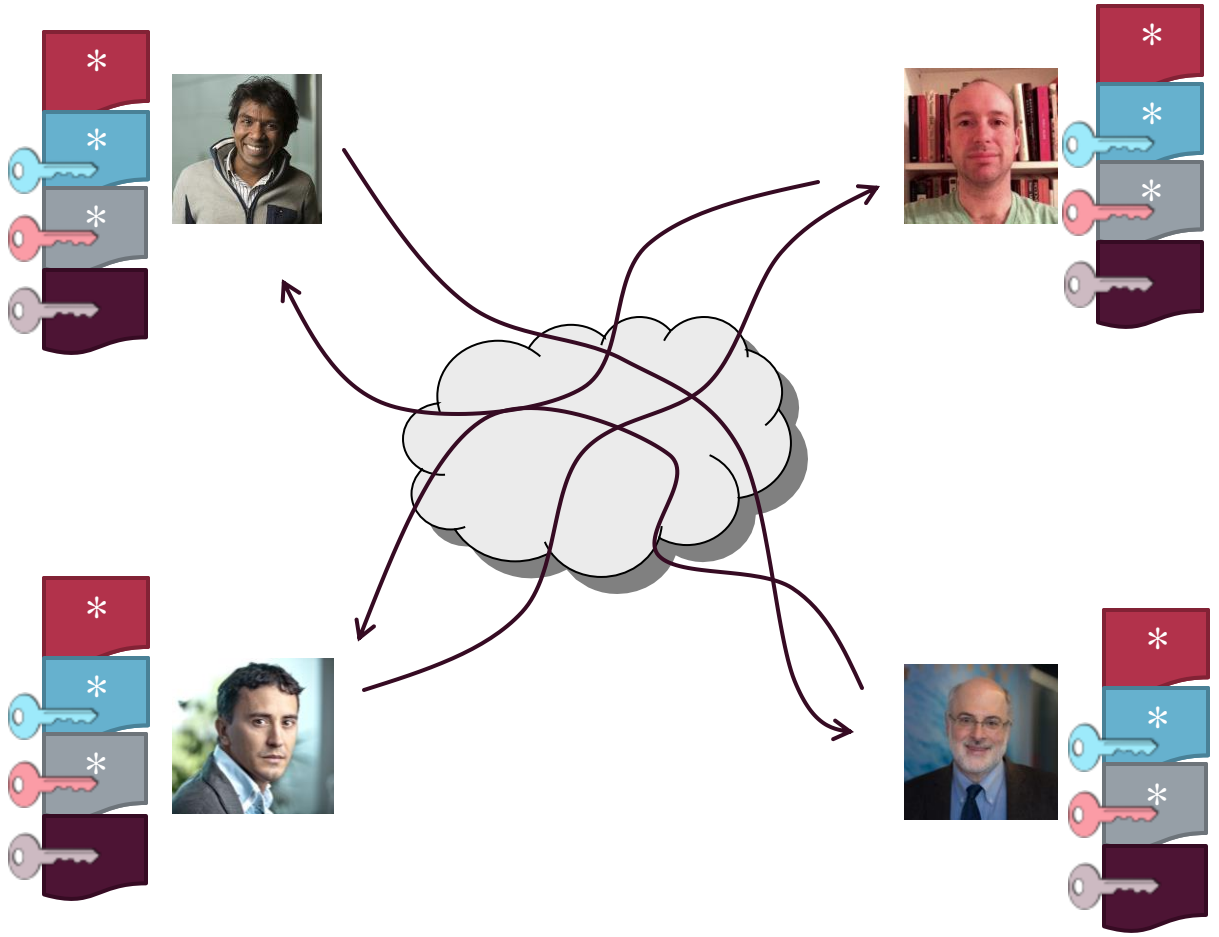
EVERYBODY SENDS LOTS OF MESSAGES TO EVERYONE





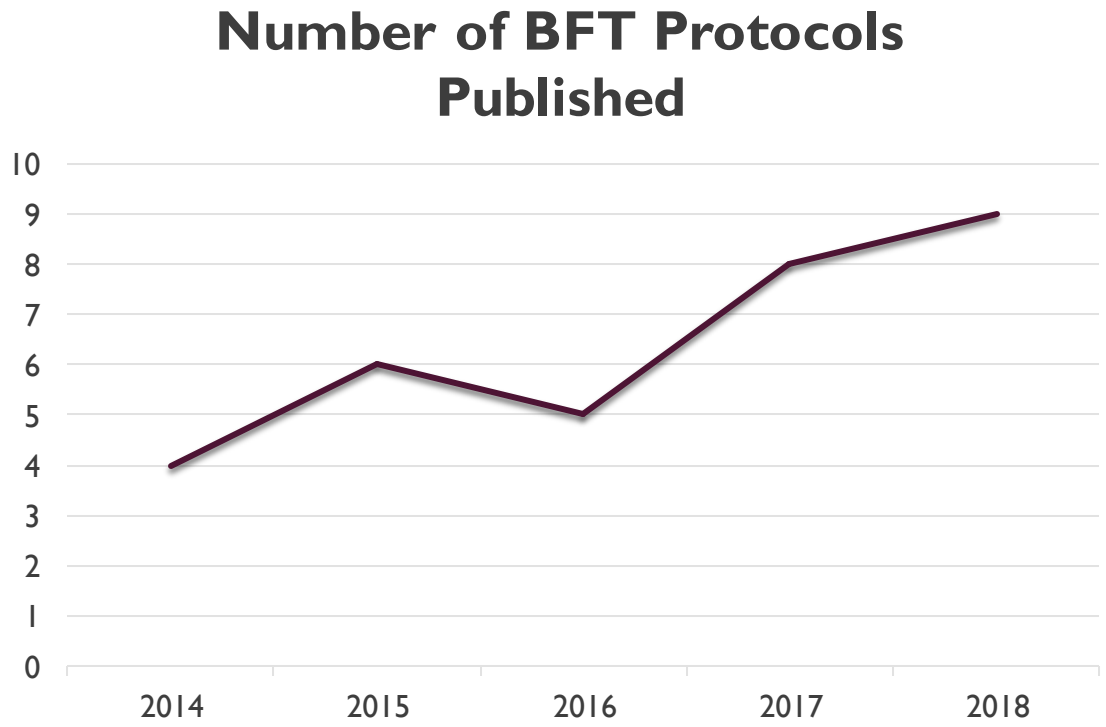
HOPEFULLY THIS CONVERGES EVENTUALLY

- Everyone will get the same set of proposals, votes, criticism, etc.
- If enough researchers agree on each step, you can make progress
- But there's no guarantee they will agree
- Oh well, BFT protocols aren't live anyway



EVALUATION

- Somehow, this usually works in practice
- Many papers on BFT algorithms have been written collaboratively



I HOPE YOU DON'T HAVE ANY QUESTIONS