REMINDER

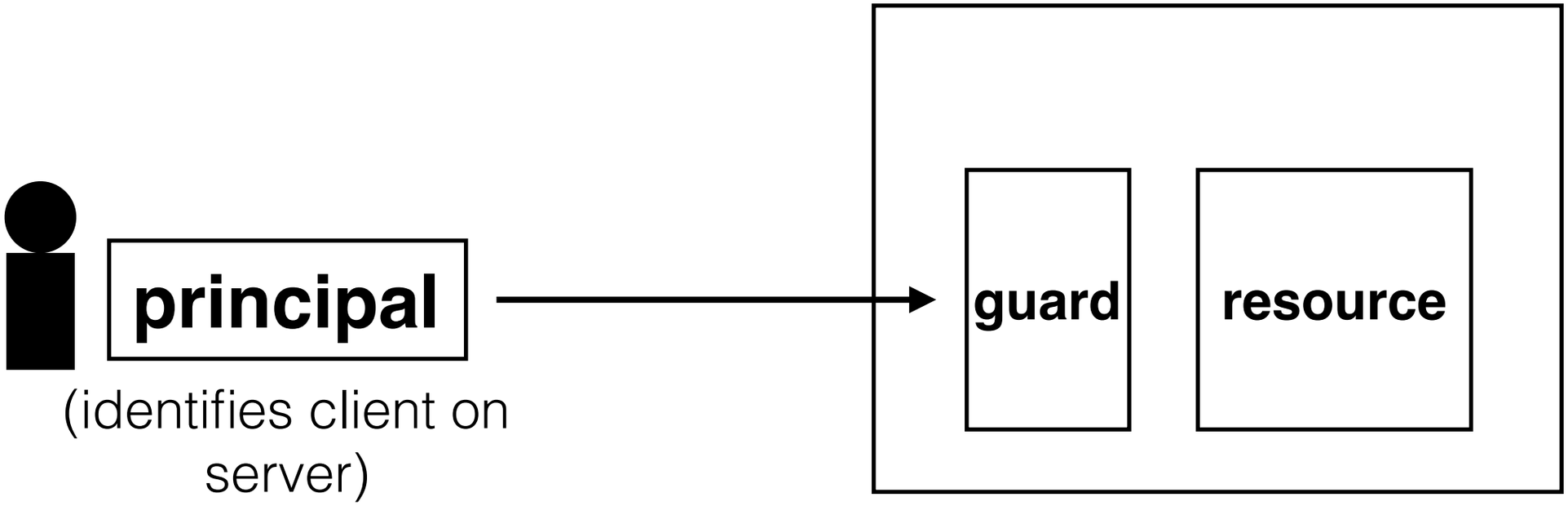
course evaluations are online
http://web.mit.edu/subjectevaluation/

please fill them out — they provide extremely valuable feedback to all instructors

6.033 Spring 2019

Lecture #24

- **Bitcoin and blockchains (and anonymity)**


principal
(identifies client on server)
guard
resource

suppose we want a distributed (and decentralized) public log that multiple users can read from/write to

one motivation: digital currency

challenge: what if I don't want other users to be able to tie my transactions back to me personally?

e.g., I don't want everyone in the world to know that Katrina LaCurts just spent money buying 10,000 tiny hands

suppose we want a distributed (and decentralized) public log that multiple users can read from/write to

one motivation: digital currency

users will be anonymous

identified only by their public keys

challenge: how do we prevent an adversary from editing our transactions?

suppose we want a distributed (and decentralized) public log that multiple users can read from/write to

one motivation: digital currency

users will be anonymous

identified only by their public keys

users will sign their transactions

prevents adversary from tampering with log data

challenge: how do we prevent an adversary from reordering/removing/duplicating transactions?

suppose we want a distributed (and decentralized) public log that multiple users can read from/write to

one motivation: digital currency

users will be anonymous

identified only by their public keys

users will sign their transactions

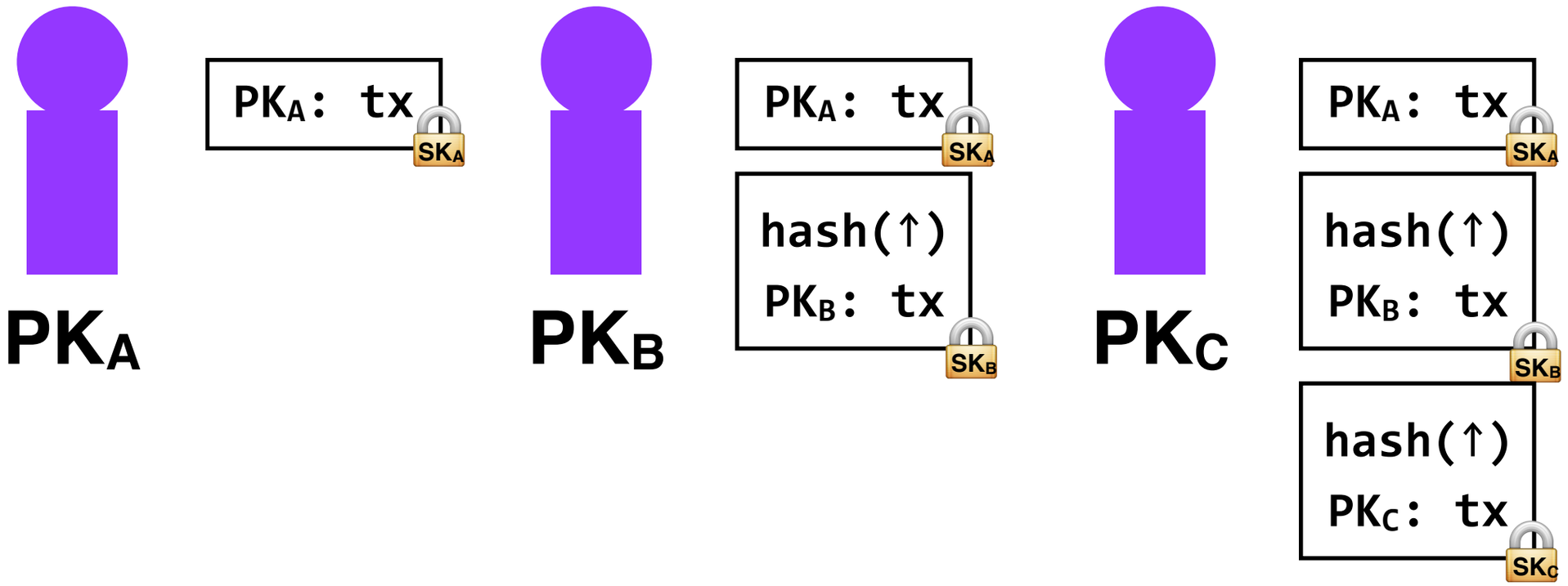
prevents adversary from tampering with log data

users will include a hash of the previous transaction as part of their signed data

prevents adversary from re-ordering

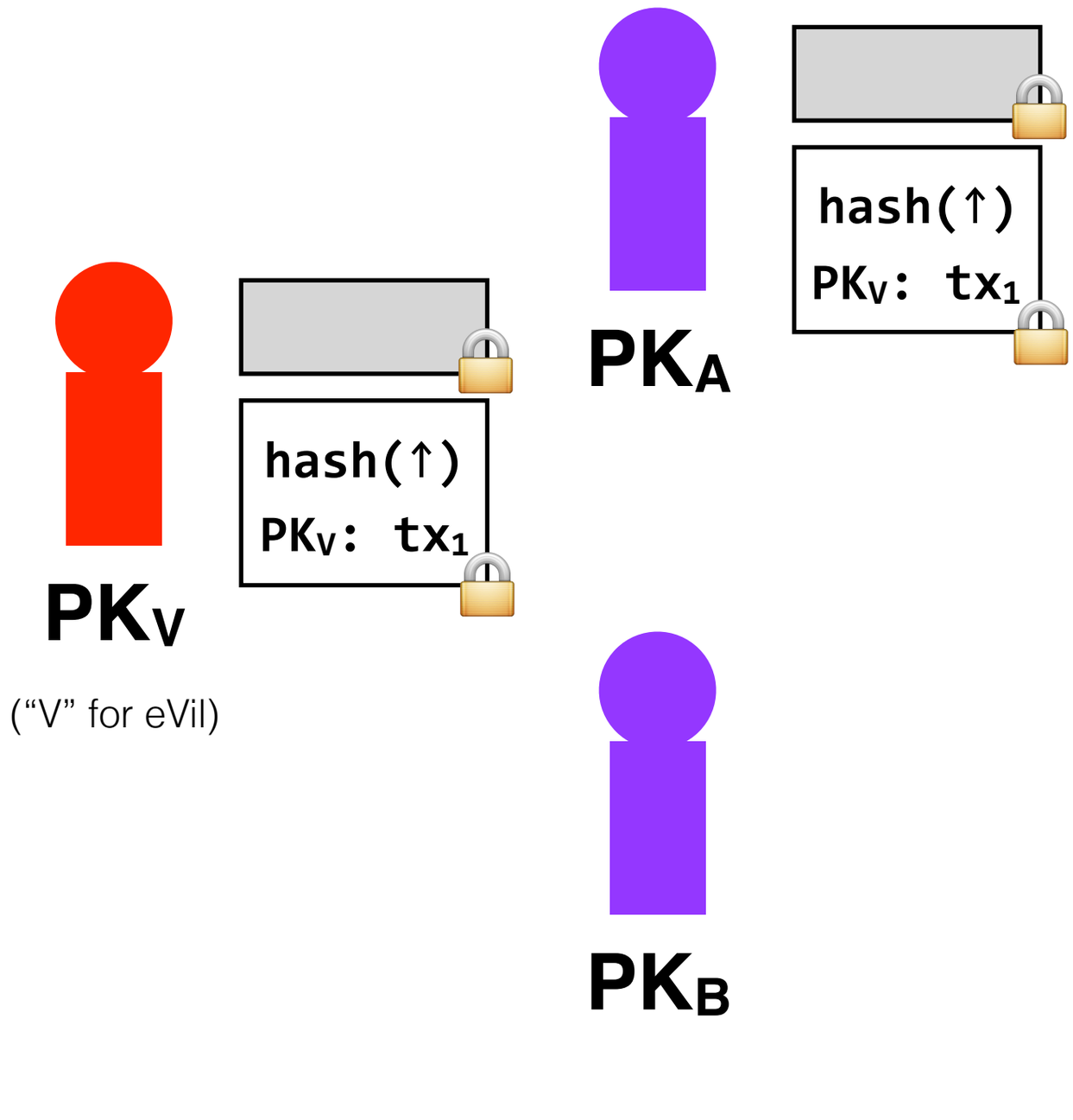
suppose we want a distributed (and decentralized) public log that multiple users can read from/write to

one motivation: digital currency



suppose we want a distributed (and decentralized) public log that multiple users can read from/write to

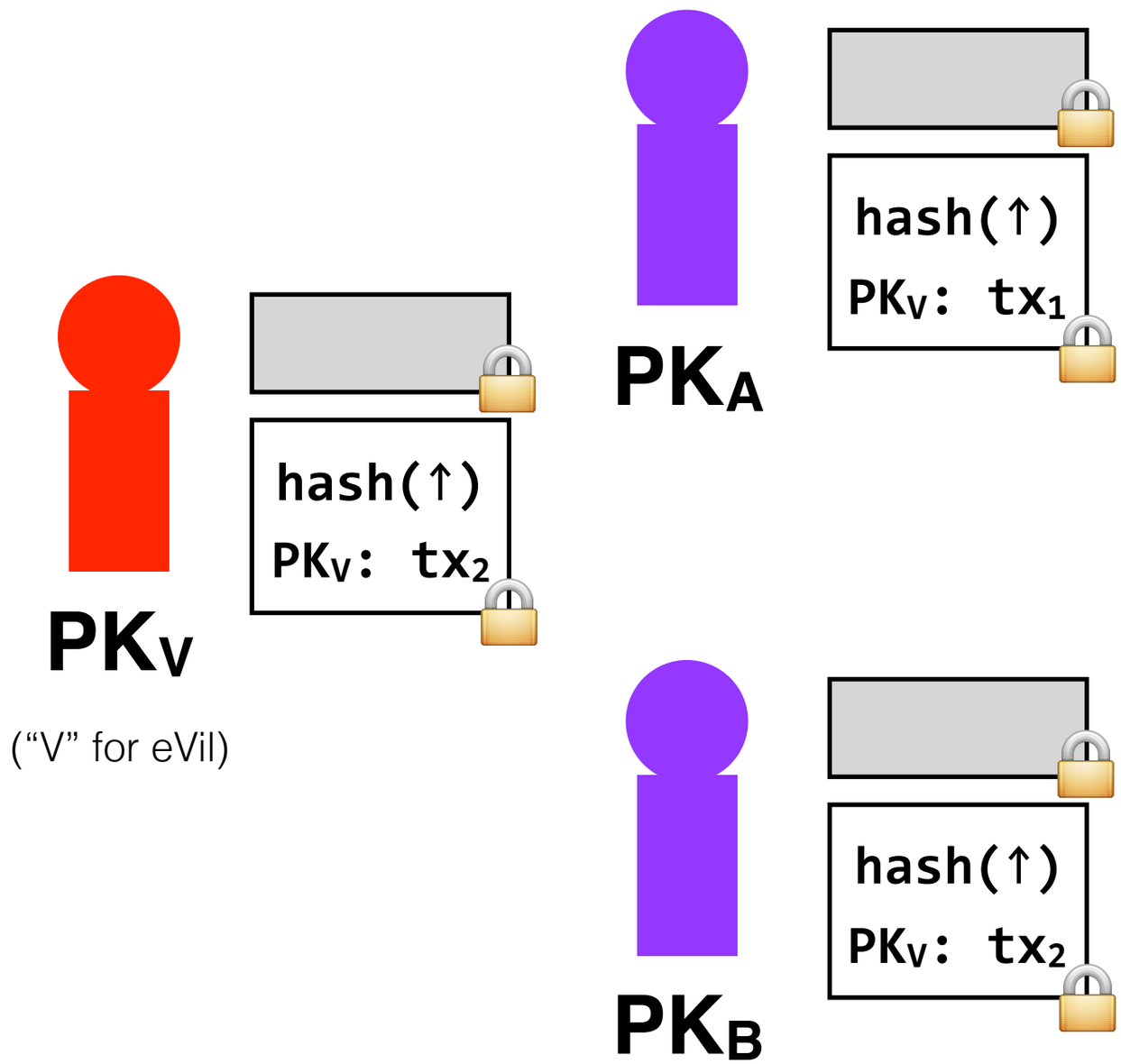
one motivation: digital currency



("V" for eVil)

suppose we want a distributed (and decentralized) public log that multiple users can read from/write to

one motivation: digital currency



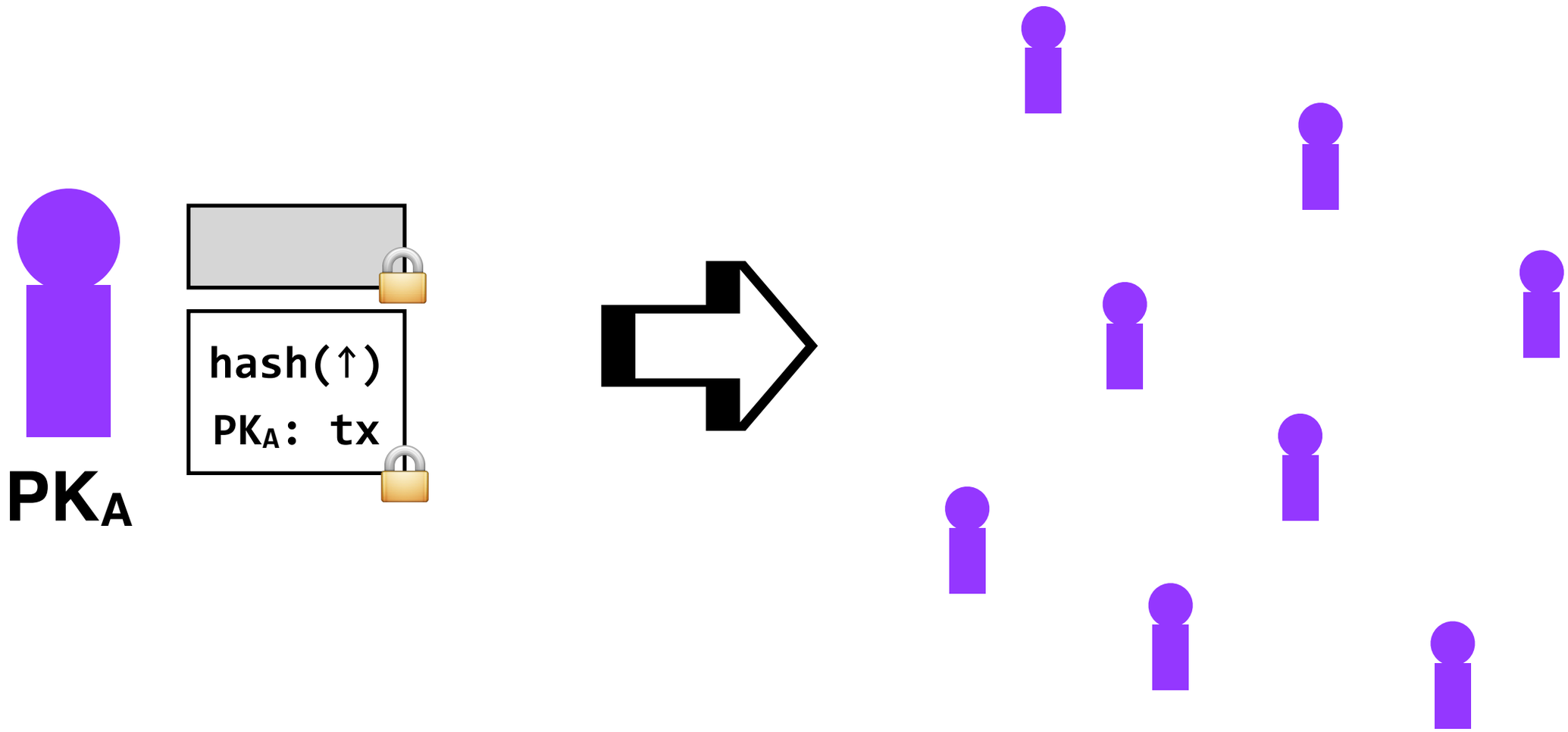
in the world of digital currency, PK_V might do this to try to spend the same "coin" in multiple places

which log is correct?

we need **consensus**

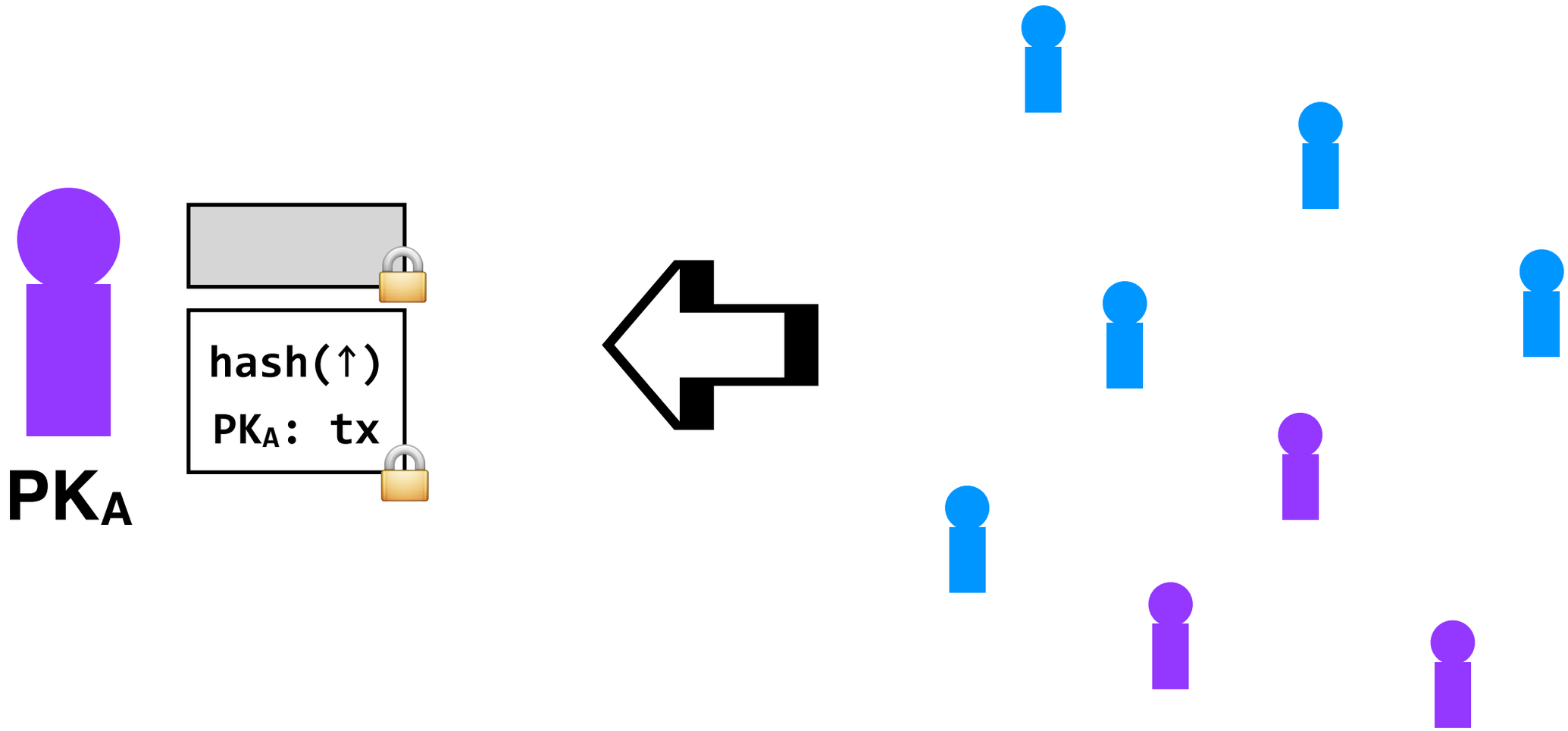
note that a similar situation could occur for completely innocuous reasons, not just because of an attack from PK_V

goal: the system should come to a consensus on what the correct log is, and ignore all other logs



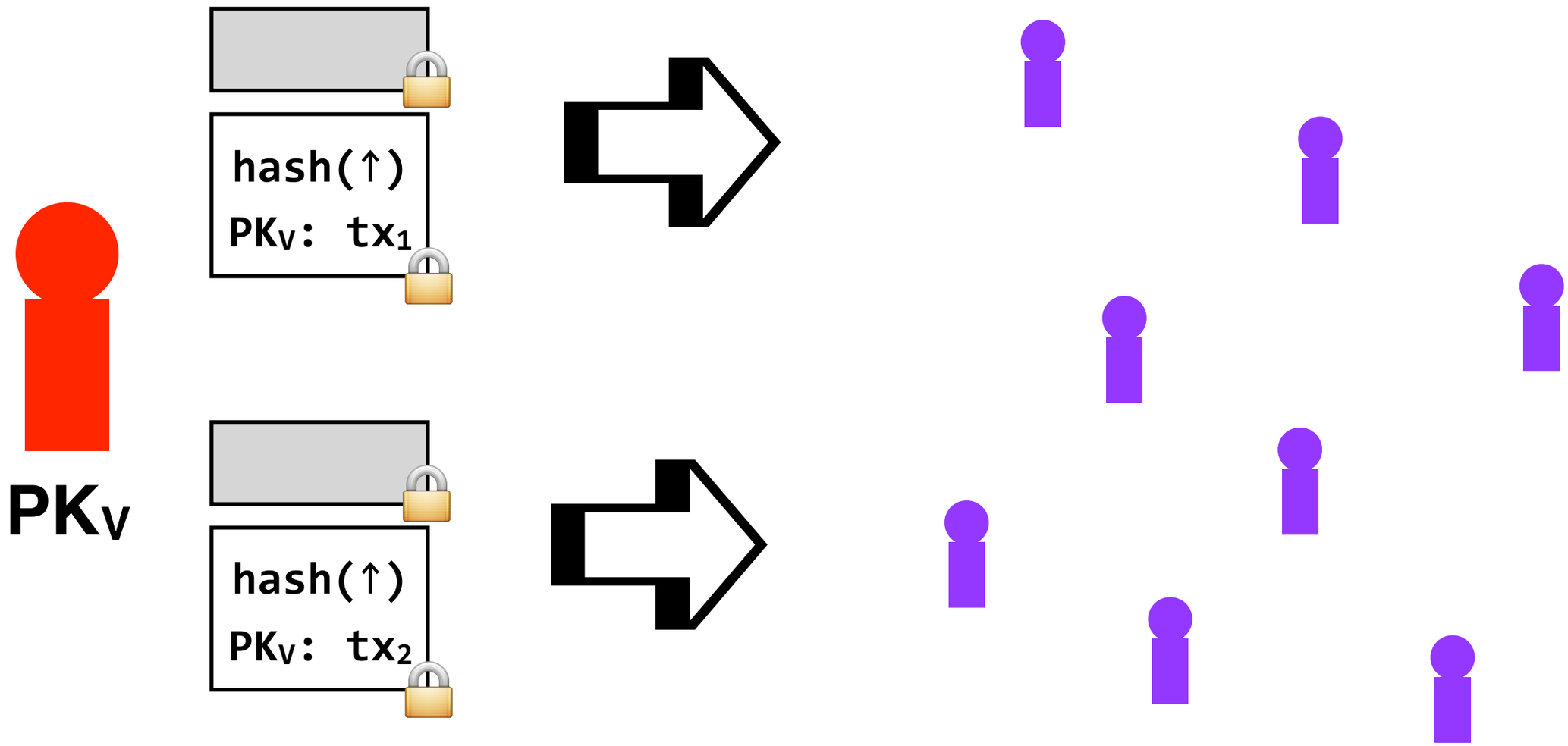
idea: PK_A broadcasts their transaction to the network, waits for a majority of users to confirm it

goal: the system should come to a consensus on what the correct log is, and ignore all other logs



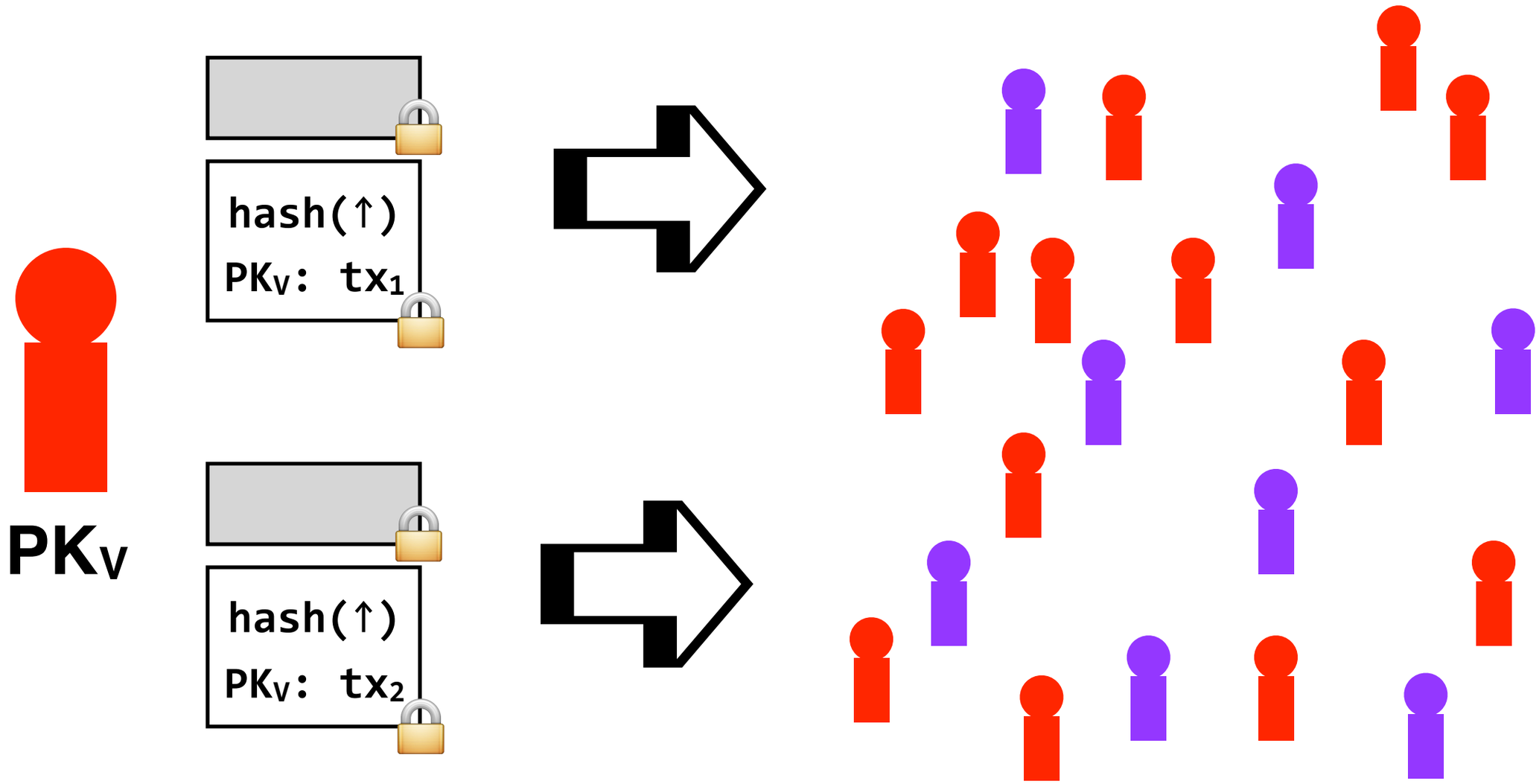
idea: PK_A broadcasts their transaction to the network, waits for a majority of users to confirm it

goal: the system should come to a consensus on what the correct log is, and ignore all other logs



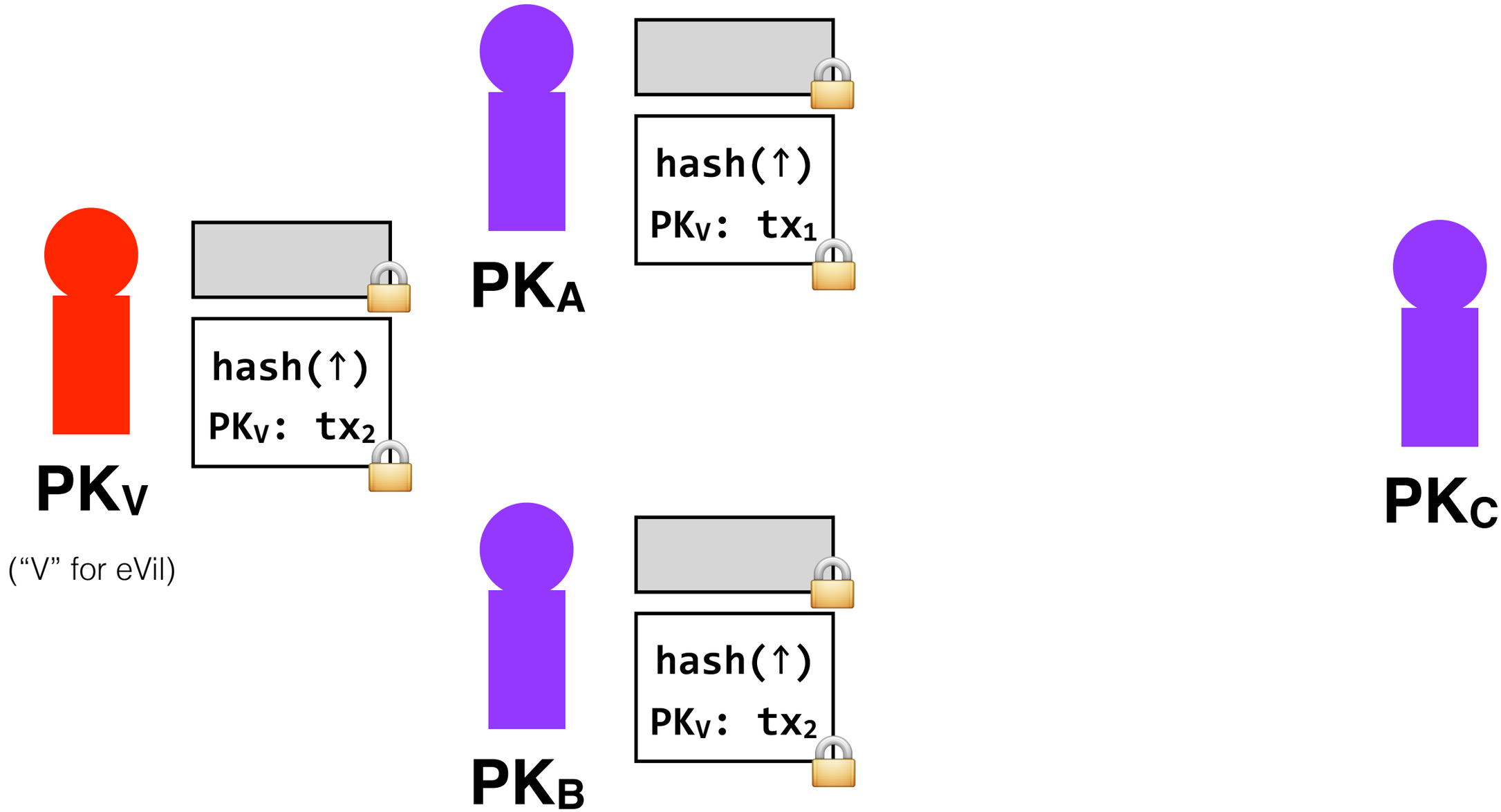
idea: PK_V broadcasts their transaction to the network, waits for a majority of users to confirm it

goal: the system should come to a consensus on what the correct log is, and ignore all other logs



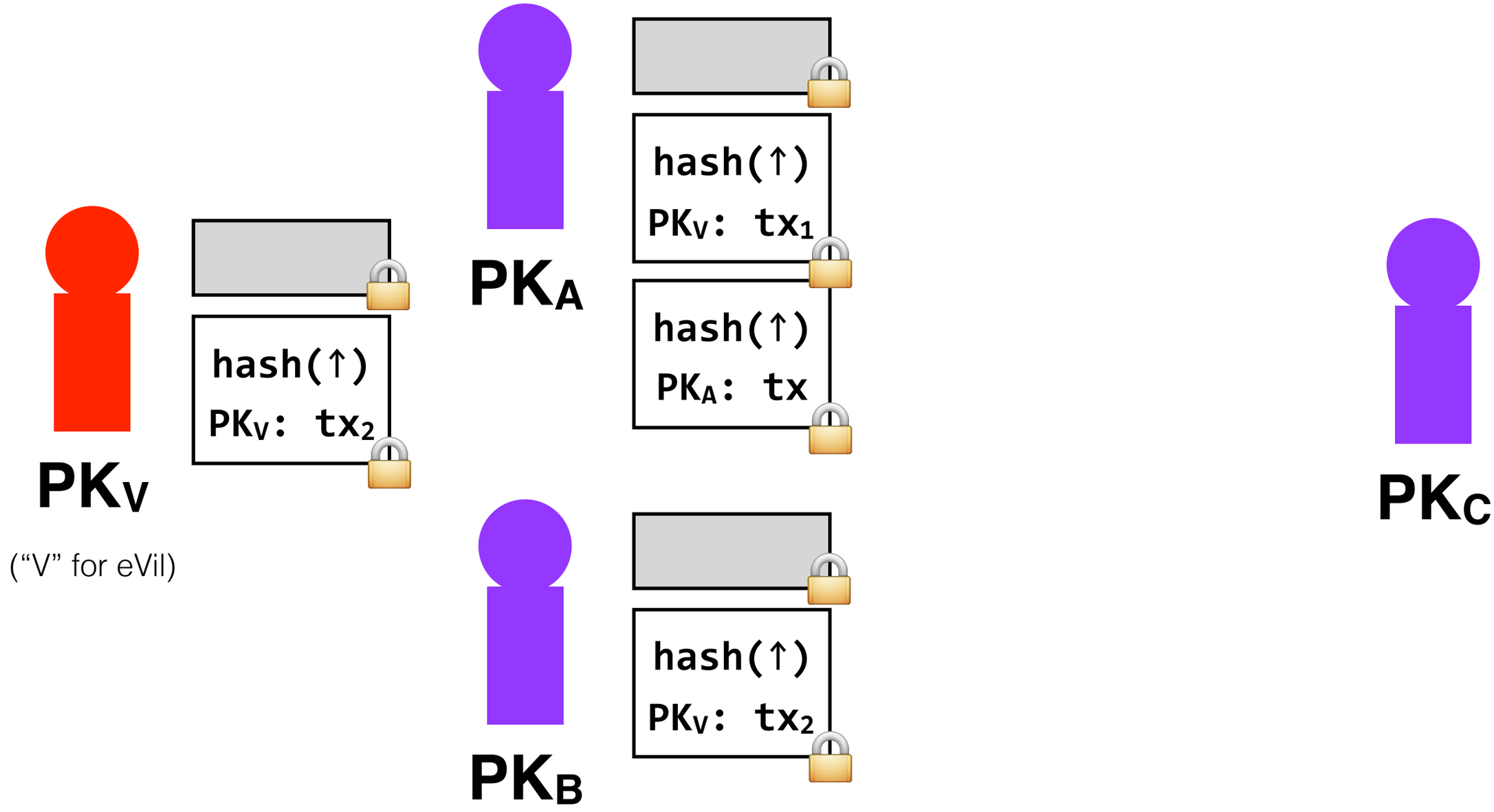
problem: PK_v can create many identities ("sybils") on the network, verify any transaction they want

goal: the system should come to a consensus on what the correct log is, and ignore all other logs



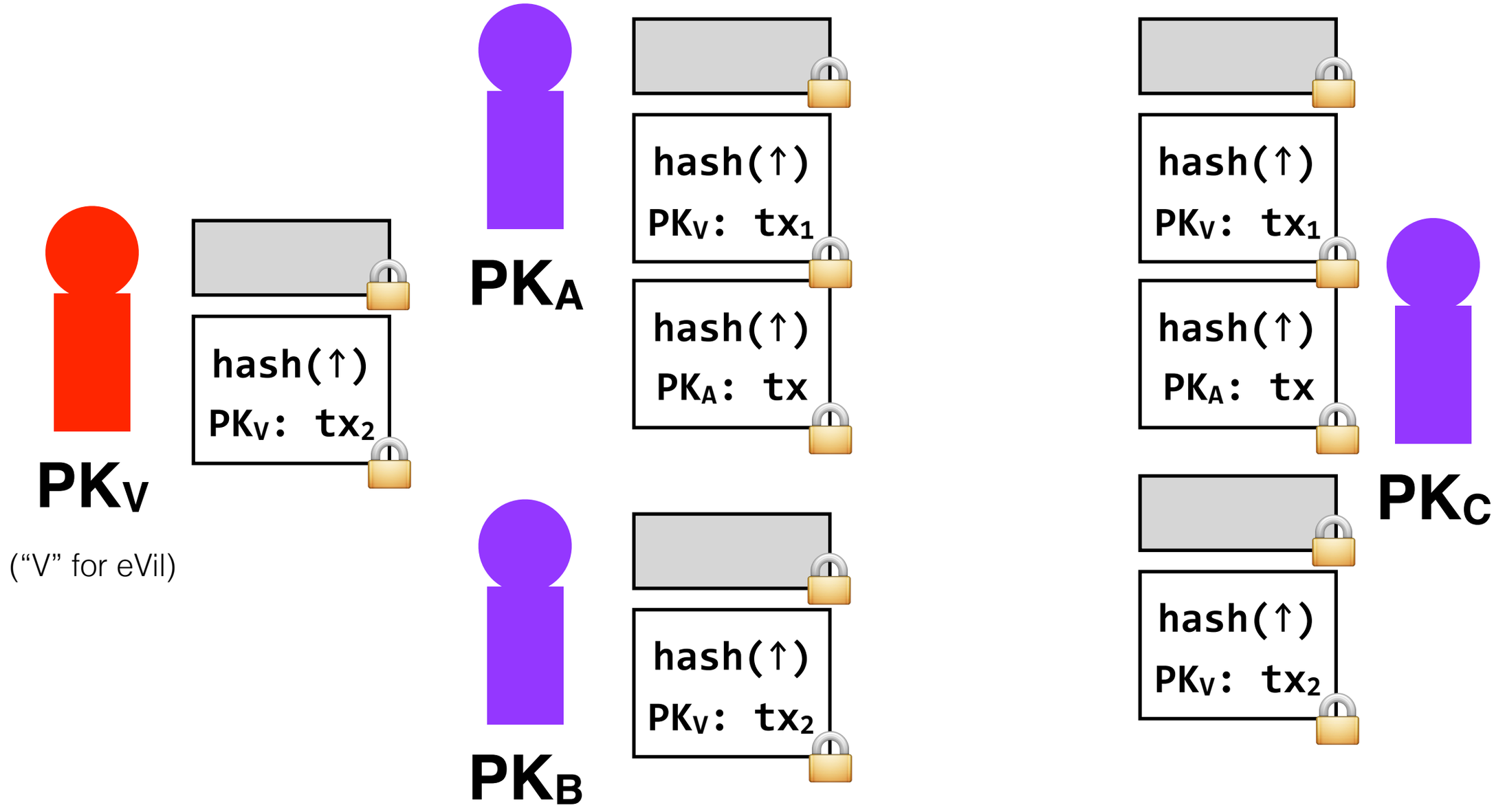
idea: if users receive different logs, they only append to the one that is longest, and don't distribute any others

goal: the system should come to a consensus on what the correct log is, and ignore all other logs



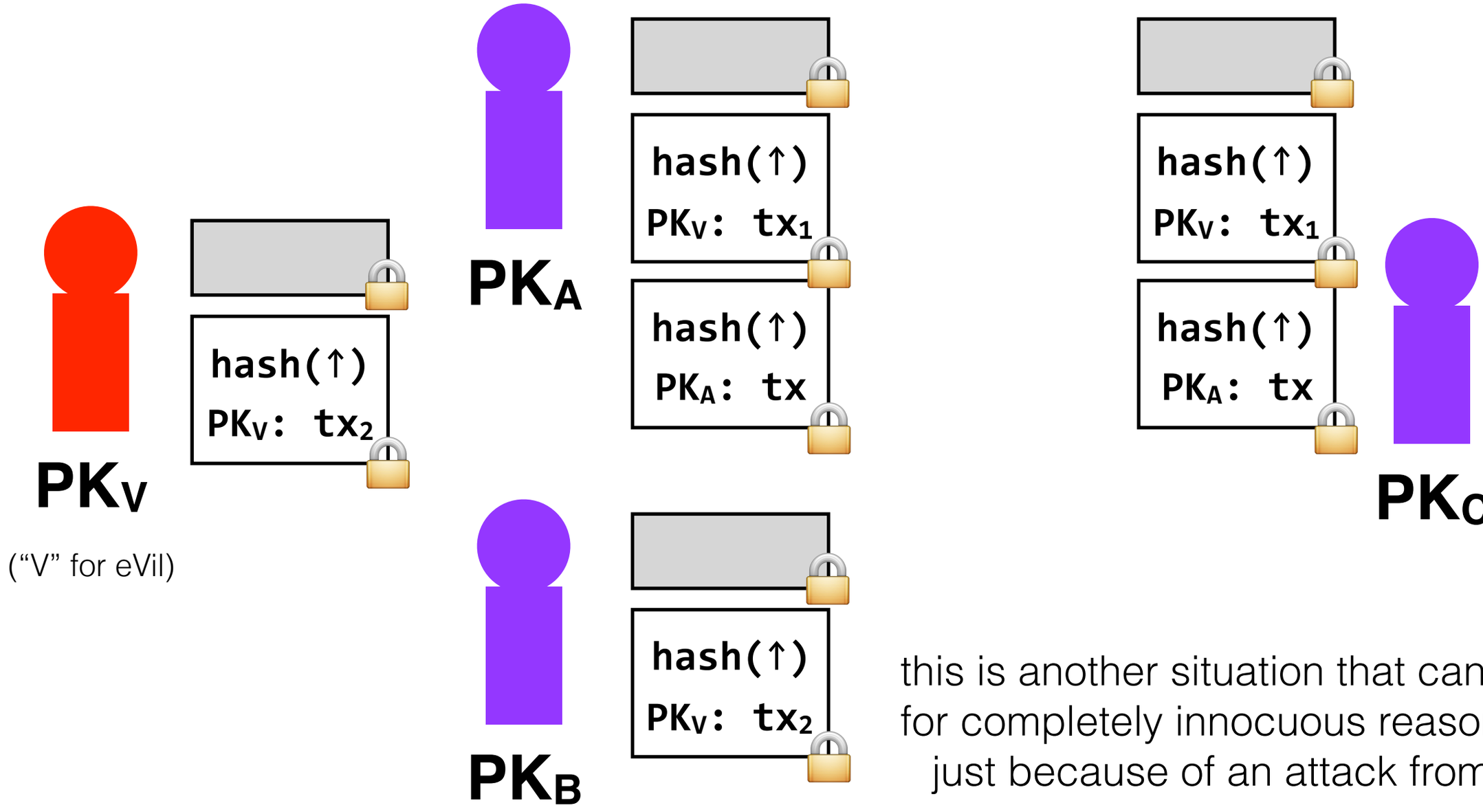
idea: if users receive different logs, they only append to the one that is longest, and don't distribute any others

goal: the system should come to a consensus on what the correct log is, and ignore all other logs



idea: if users receive different logs, they only append to the one that is longest, and don't distribute any others

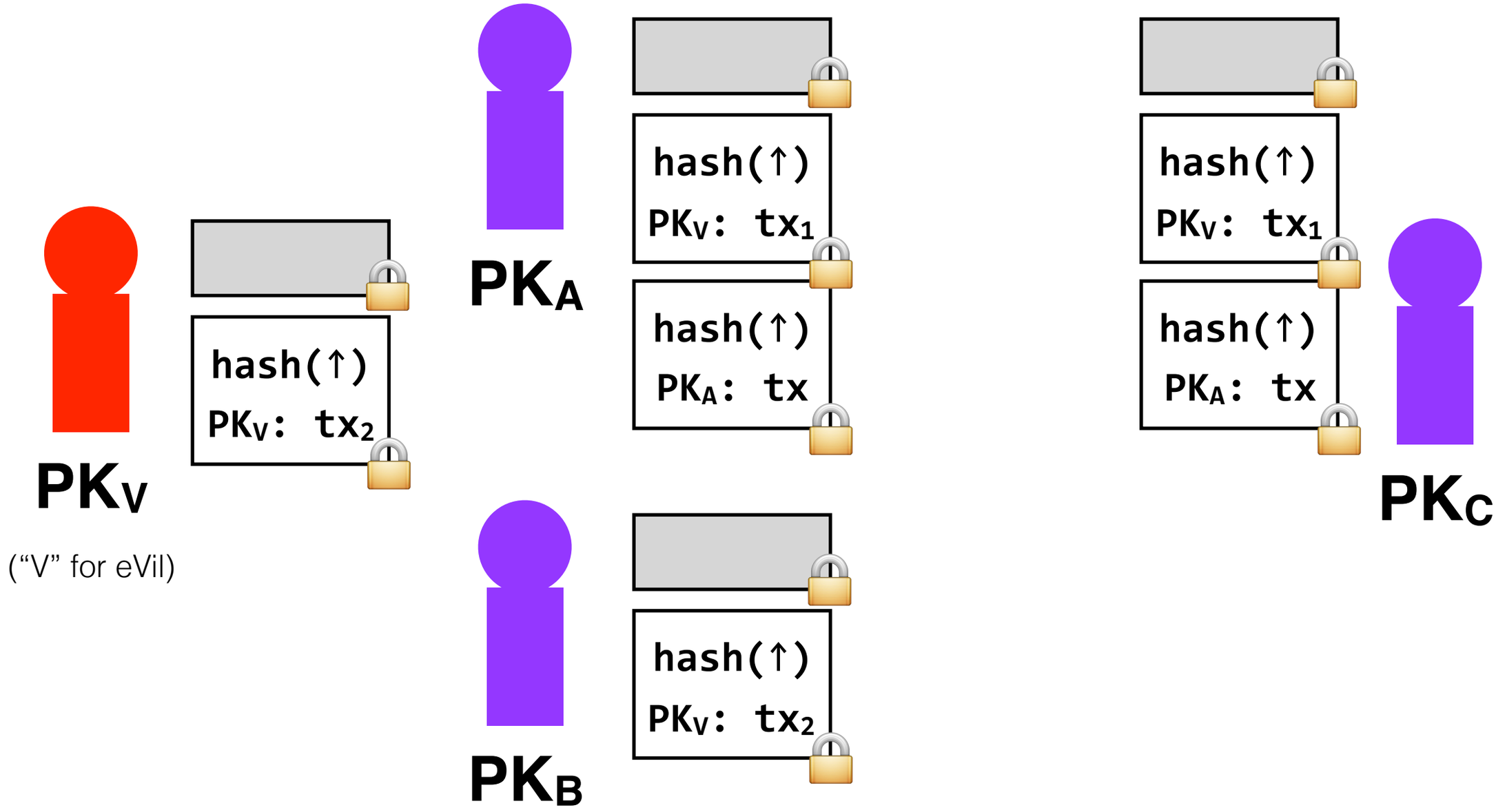
goal: the system should come to a consensus on what the correct log is, and ignore all other logs



this is another situation that can occur for completely innocuous reasons, not just because of an attack from PK_V

idea: if users receive different logs, they only append to the one that is longest, and don't distribute any others

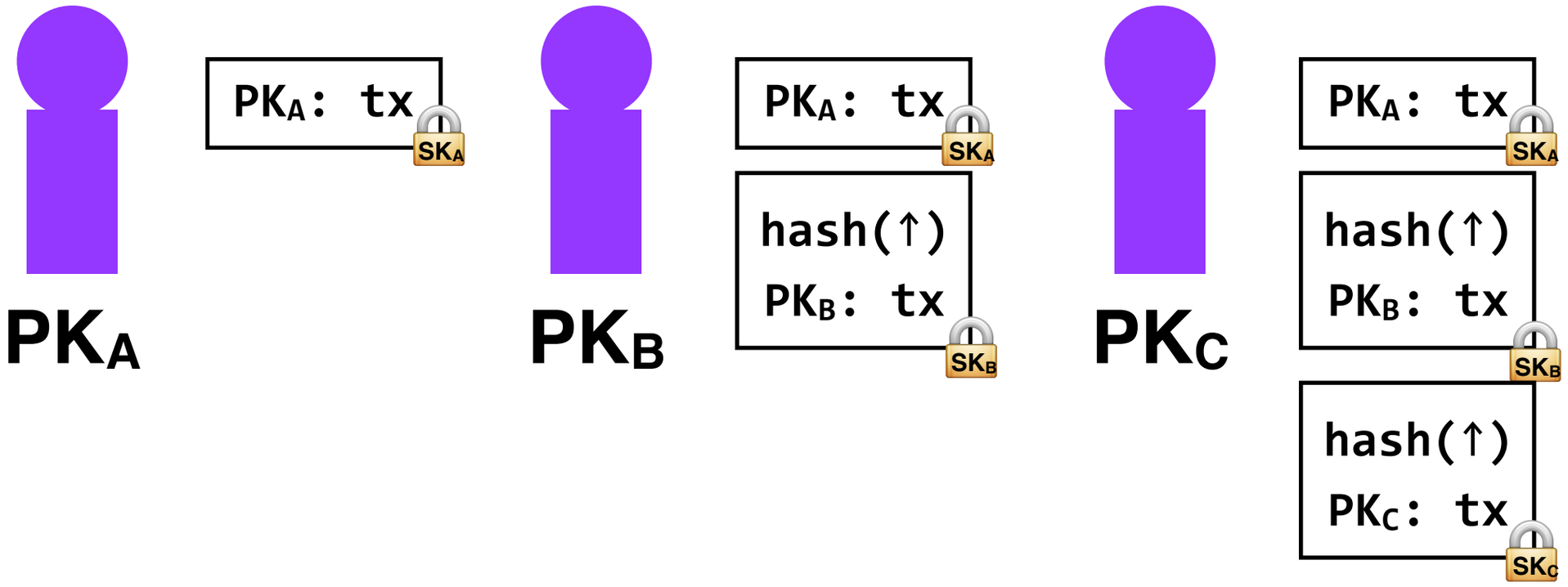
goal: the system should come to a consensus on what the correct log is, and ignore all other logs



problem: at what point can a user decide that their transaction is confirmed?

e.g., how does PK_A know that their transaction is part of the "correct" log?

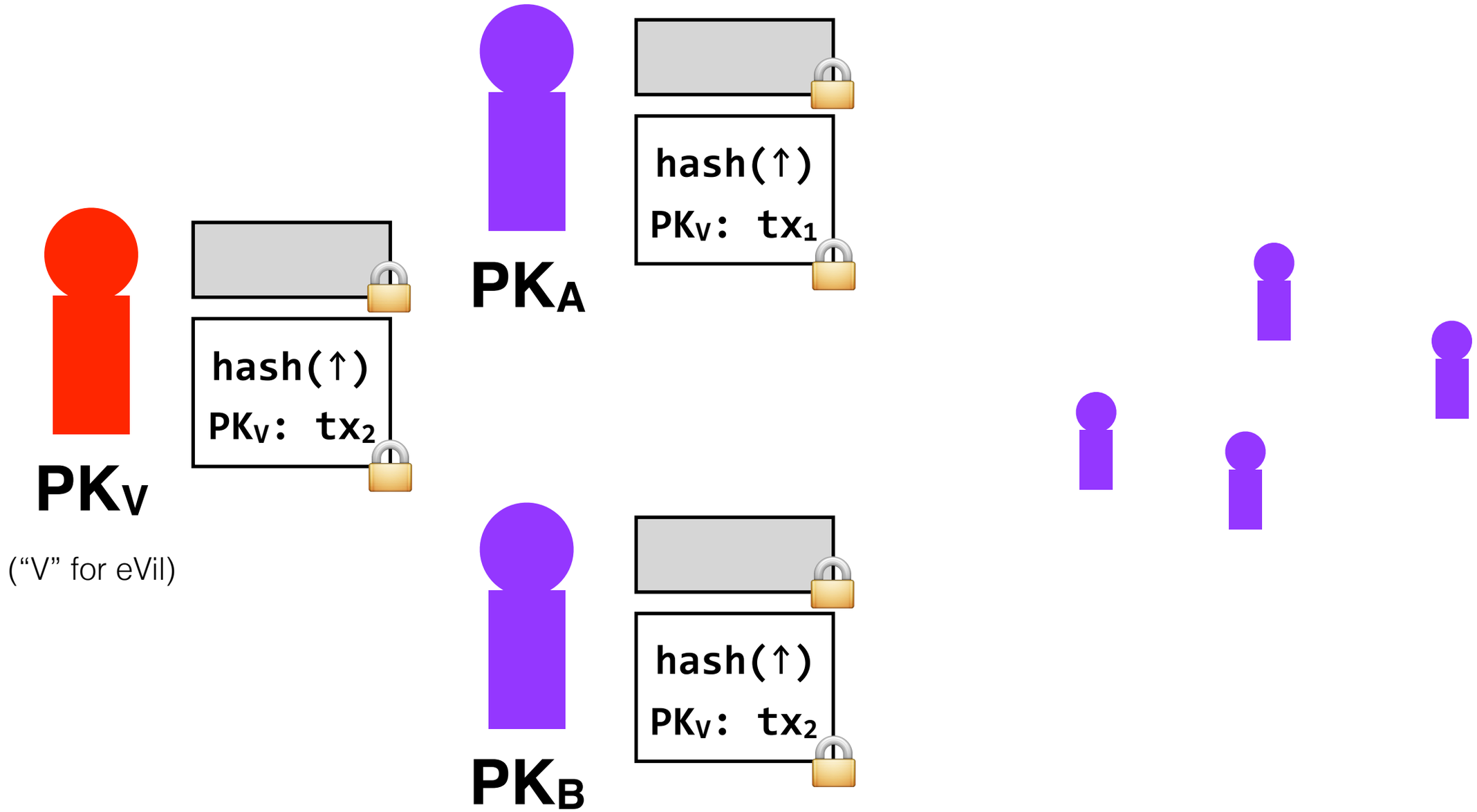
goal: the system should come to a consensus on what the correct log is, and ignore all other logs



if N = 2, PK_A's transaction is confirmed at this point

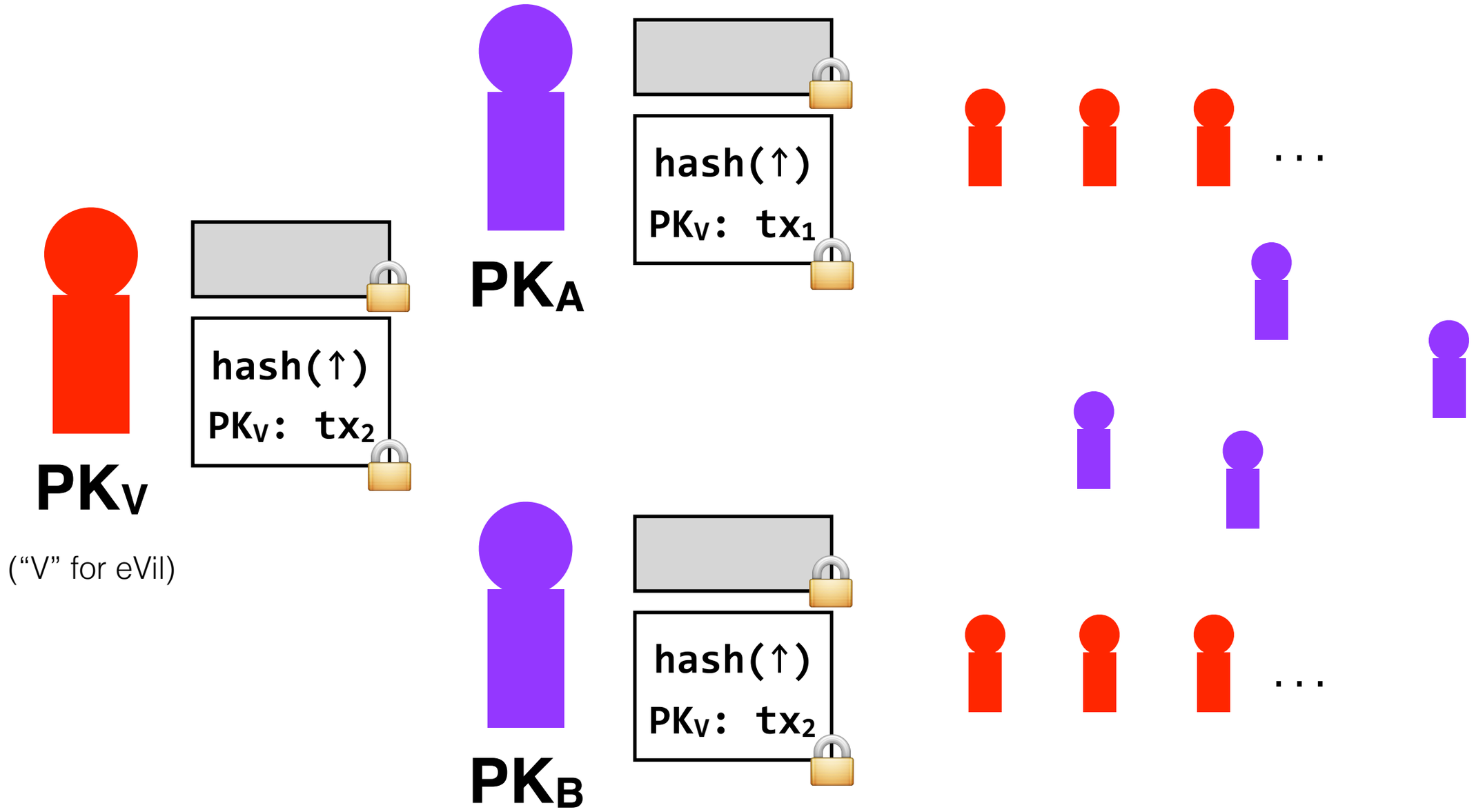
idea: transaction t is confirmed when it's followed by N blocks in the log

goal: the system should come to a consensus on what the correct log is, and ignore all other logs



idea: transaction t is confirmed when it's followed by N blocks in the log

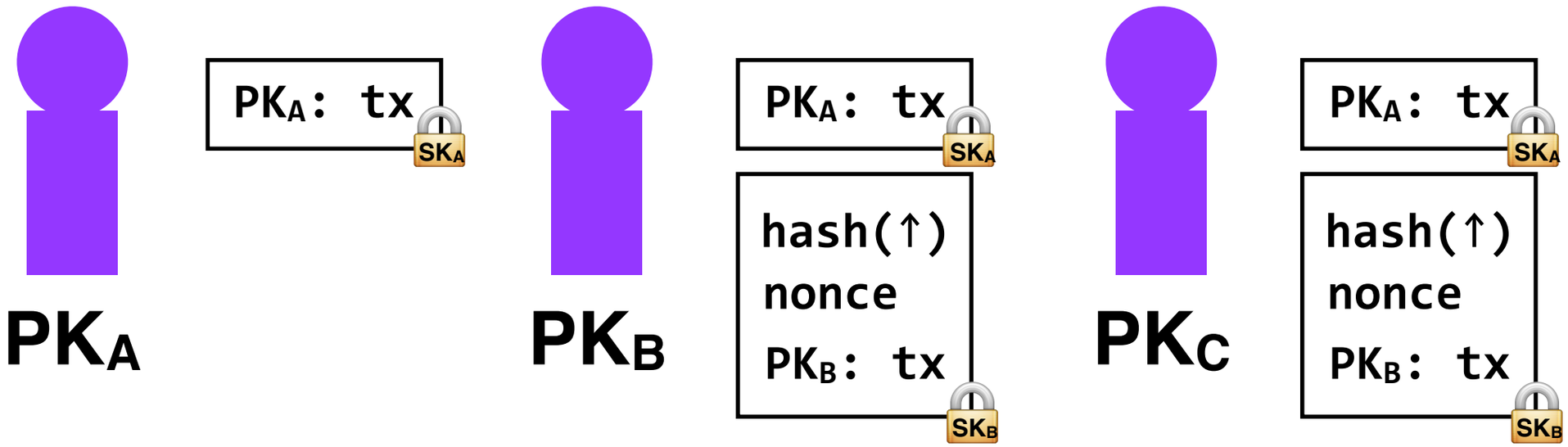
goal: the system should come to a consensus on what the correct log is, and ignore all other logs



problem: PK_V can create sybil and validate as many blocks as is necessary; it's *easy* to validate blocks

goal: the system should make it difficult to validate blocks

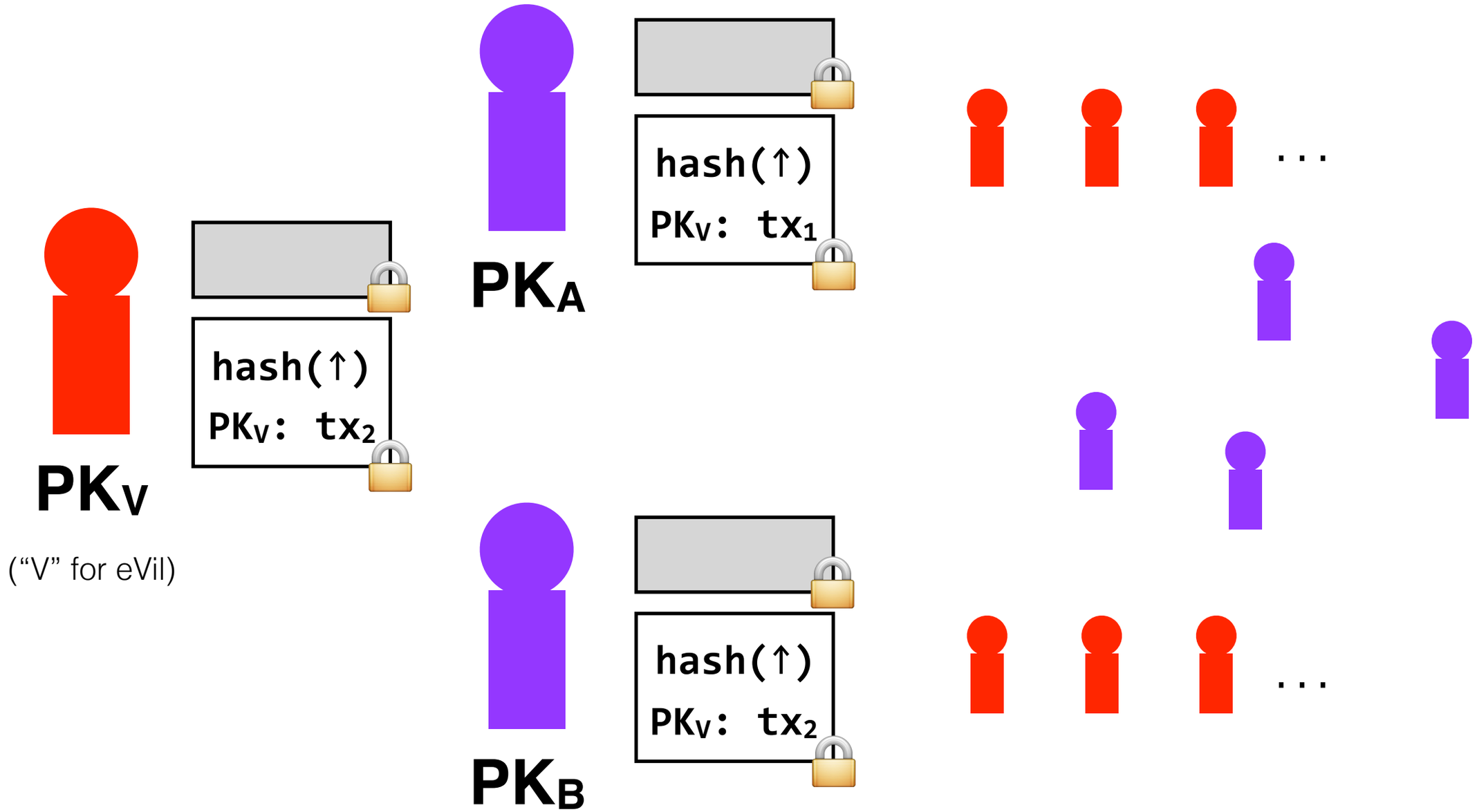
(validate = add a block to the log)



solution: proofs of work

goal: the system should make it difficult to validate blocks

(validate = add a block to the log)

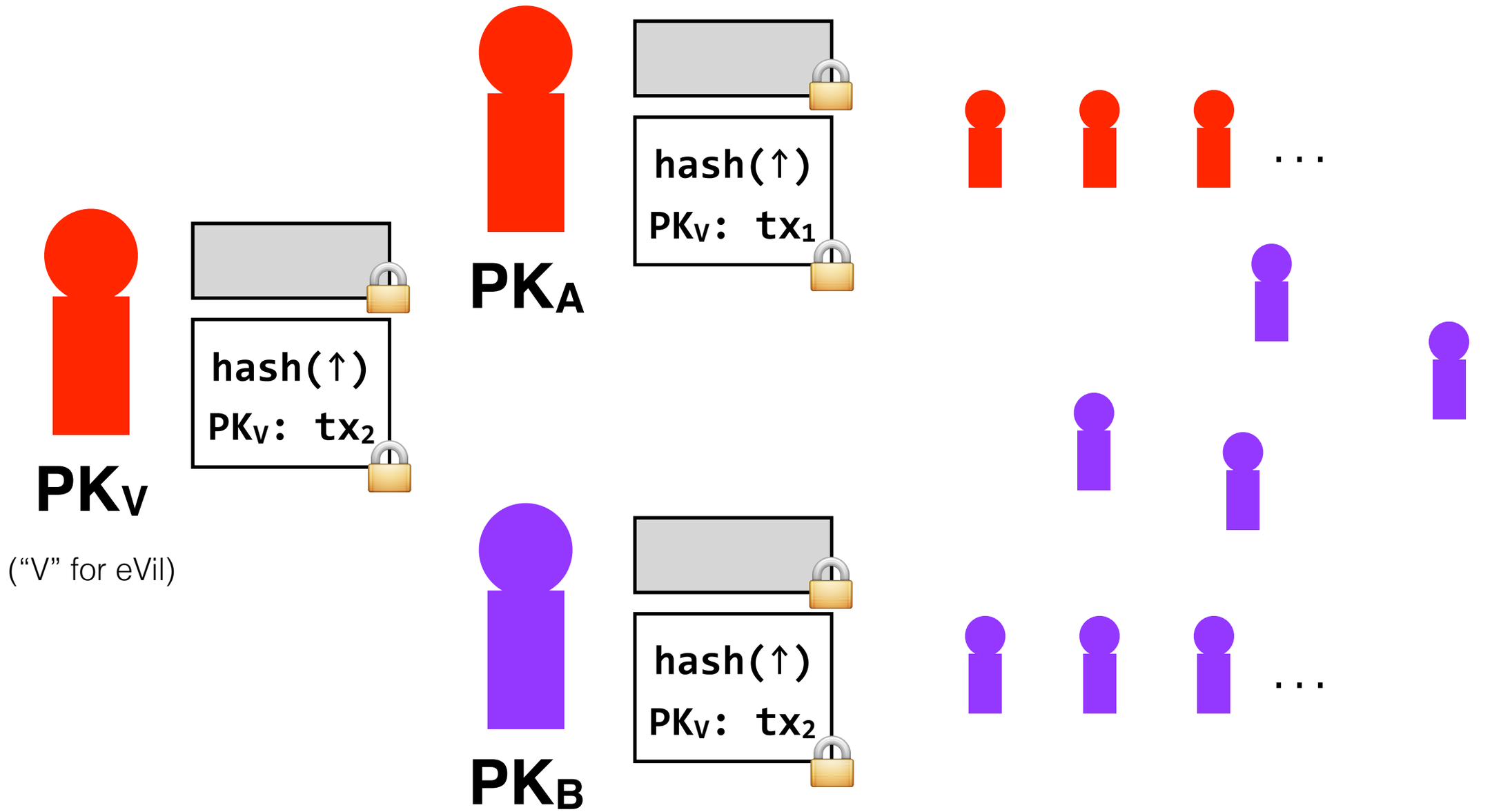


now it is difficult for PK_V to validate all of the necessary blocks, even with Sybils

(requires a lot of compute power, not just a lot of identities)

goal: the system should make it difficult to validate blocks

(validate = add a block to the log)

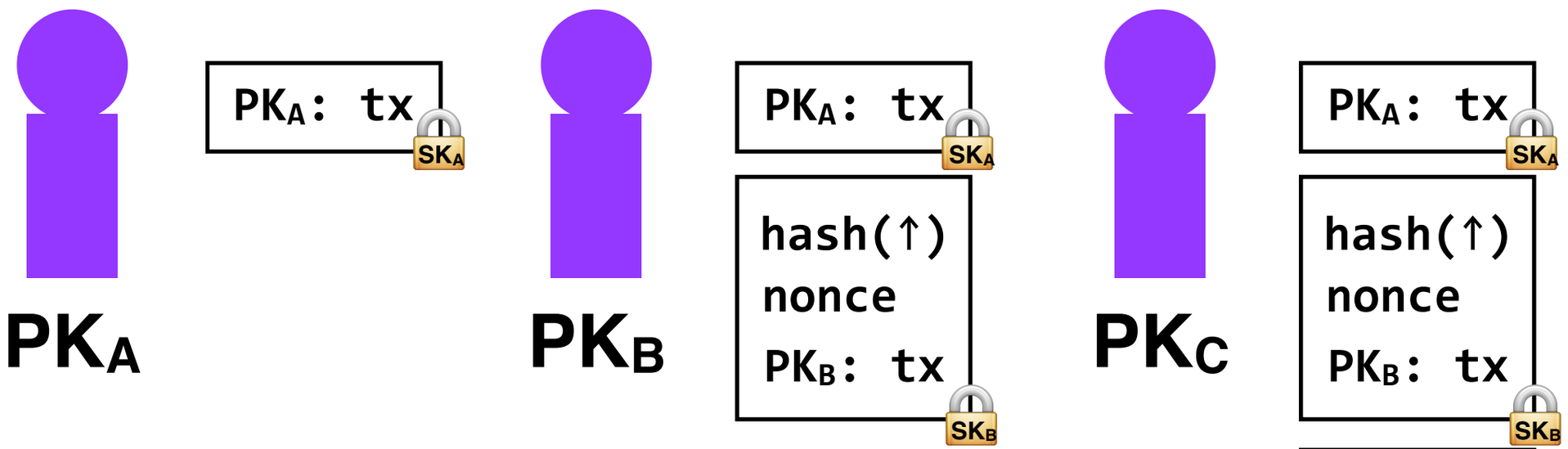


now it is difficult for PK_V to validate all of the necessary blocks, even with Sybils

(requires a lot of compute power, not just a lot of identities)

a distributed public log that multiple users can read from/write to

one motivation: digital currency

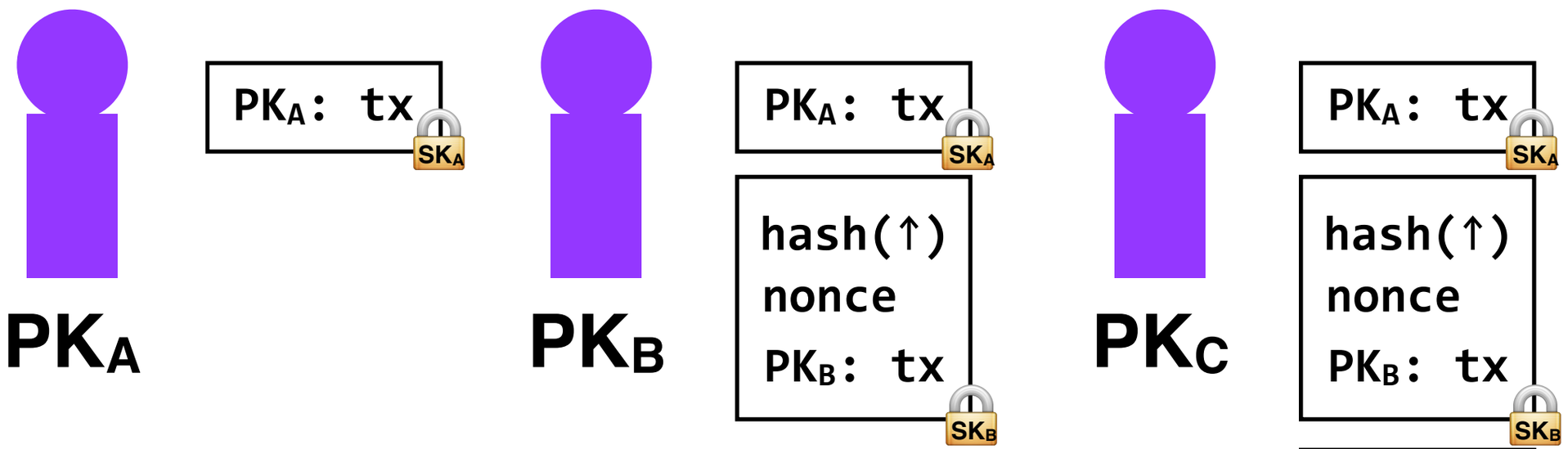


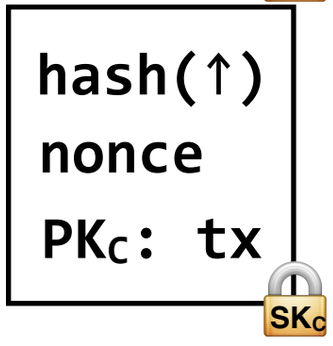
this log is the blockchain

(typically a block will contain multiple transactions, not one as we've been showing)

a distributed public log that multiple users can read from/write to

one motivation: digital currency



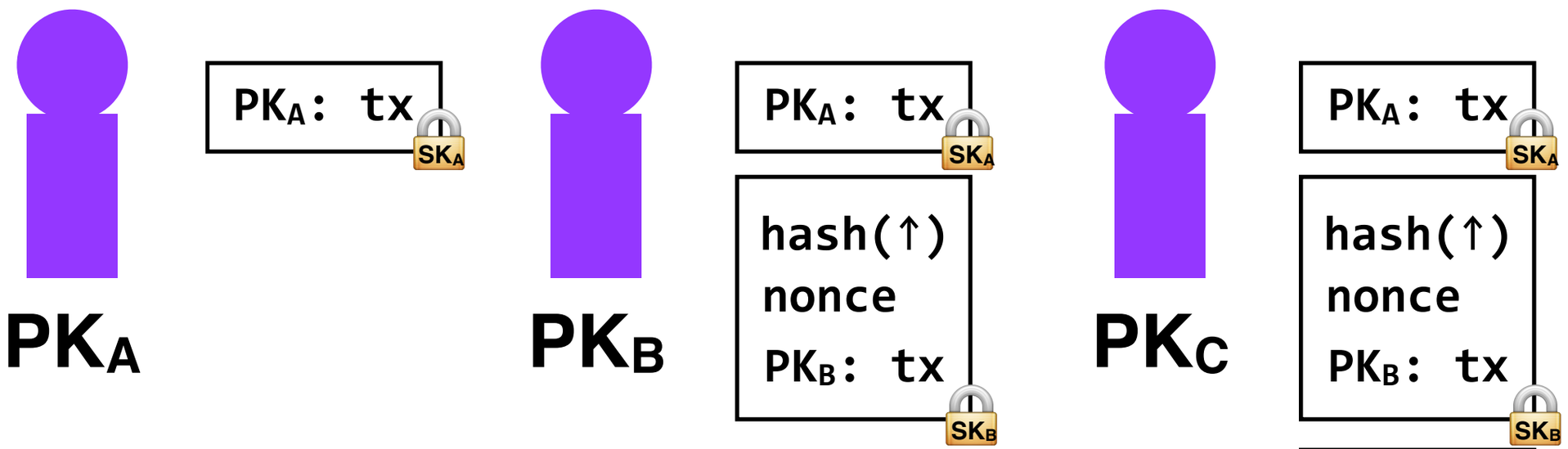


the process of validating transactions is known as mining

users receive payment (bitcoins) for validating blocks

a distributed public log that multiple users can read from/write to

one motivation: digital currency

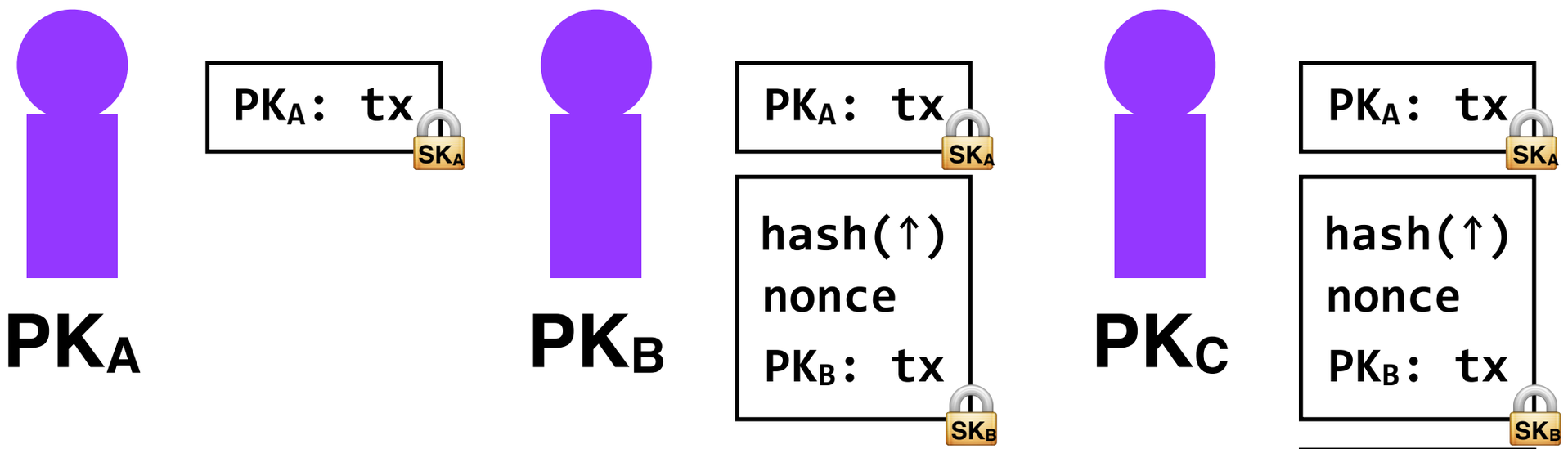


if multiple users validate the same block at (roughly) the same time, the blockchain will fork

the system resolves this problem since miners will only work on the longest fork

a distributed public log that multiple users can read from/write to

one motivation: digital currency



a block is confirmed after 5* valid blocks follow it in the blockchain

* 5 is common, but this is a tunable parameter

- **Bitcoin** is a decentralized digital currency. Being decentralized means that there is no bank; in Bitcoin, everyone is the bank.

- Bitcoin provides a distributed public log called the **blockchain** that can be used for purposes other than digital currency. It uses **proofs-of-work** to prevent Sybil Attacks, since strong identities won't work.

- In theory, users of Bitcoin are **anonymous**; in practice, it's not clear how true that is.