CryptoKupa: Decentralized Communal Funds with Bitcoin

Ori Shimony

University of Virginia <os7hf@virginia.edu>

Abstract. There are very few ways for communities to share money in a secure and practical manner. Doing so often requires high levels of trust and sustained physical proximity among members. I propose a scheme that allows a group of individuals to create and operate a shared Bitcoin fund without a central authority. Contributions to the fund are made using a multisig transaction scheme that distributes control amongst members equally. This ties the difficulty of spending to the size of the withdrawal, as more peers are required to approve larger amounts. The system's parameters also make it flexible to the needs of its users.

1 Background

CryptoKupa was motivated by my experiences in Habonim Dror, an international progressive labor Zionist youth movement. For ten years I spent my summers at camp Moshava, one of the movement's seven North American summer camps, including two summers on staff. Amongst other goals, the movement aims to participate in the creation of a social order based on the principles of self-determination, individual freedom, political democracy, cooperative economics, and the equality of human value[1]. In practice these aspirations manifests through educational efforts to create activists dedicated to social justice, equality, and peace.

Members utilize cooperative living frameworks as the foundation for actualizing these goals and for concretely expressing their values. As a group of fifty counselors working around the clock at Moshava for eight weeks, we formed our own cooperative with collectively agreed upon systems, expectations, and rules. These guided us in living amongst ourselves and in operating a summer camp while implementing an effective educational curriculum for youth aged 8 to 15. A more encompassing example of Habonim collectives is when small groups of young adults immigrate to Israel together. There they live together, using their unit as a tool of support and empowerment in pursuing social-justice related work.

These collectives operate on a system of property sharing, modeled off the Israeli Kibbutz Movement, called *Kupa*. *Kupa* (קופה) is a Hebrew word that roughly translates to cash-box. More generally it describes a system of sharing in which a group of people collectively control a communal pot of money. The term refers both to the actual pot itself and the system as a whole. A community establishes a *Kupa* through a meeting in which members decide on the rules and scope of their particular system. The group can specify parameters such as individual spending limits, suggested contribution amounts, or specifications of acceptable uses. For instance, as Moshava staff we decided that none of us would use personal money for the duration of the summer. From then on all expenses, from personal transportation costs or toiletry replacements to group dinners on nights off, were on *Kupa*. Longer-term communities, like immigrant collectives in Israel, can rely even more completely on *Kupa*, while less encompassing contexts like college housemates might construct their system to cover basic living needs only. Groups also elect a treasurer to collect, maintain, and distribute funds. In order to contribute, members give cash or some cash equivalent to the treasurer. Participants are expected to contribute reasonable amounts based on what they can afford and to spend reasonably based on their needs. To use the fund individuals request cash from the treasurer for an acceptable use or spend personal money and then submit a receipt for cash reimbursement.

The effect of *Kupa* on community is profound. Its "give what you can, take what you need" philosophy can strengthen communities by equalizing power among participants and more tangibly tying the individual to the collective. Because an individual's spending power is independent of amount contributed, one's socioeconomic standing outside of the community does not affect their power within it. In intentionally structuring internal economics as such, *Kupa* eliminates the socially divisive and exclusory qualities that money takes on in most contexts. Not only are members confident that basic needs will be met, but they can also feel comfortable spending reasonably on certain luxuries as well. Furthermore, collectivizing ownership redefines individuals in the context of a community. Every time an individual wants to use *Kupa*, even if only for a personal need, she must consider the necessity of doing so in context of the community. The consideration of community implicit with each withdrawal encourages a communal mindset conducive to responsible decision-making. Purchases that benefit the community as a whole, such as a new couch for a shared space, are also made easier since the medium of exchange is shared in the first place. Without *Kupa* or some sort of authority it is difficult to convince each member of a group that such a purchase would personally benefit him enough to agree to split the cost. Such approaches can cause divisions among a group that lead to a less cohesive community experience. In this way, *Kupa* entangles the needs, values, and goals of the personal with the communal in a mutually beneficial manner.

However, responsible spending and adequate member contributions are not guaranteed. The system succeeds due to sustained physical proximity and continuous trust among members. Mutual trust exists in these arrangements as a result of previously established, often long-standing personal relationships. This contributes to a sense of confidence that participants will "give what they can" and only "take what they need." If participants subscribe to the ideology underlying *Kupa* and are invested in the success of the community, then they benefit more from using it responsibly than from exploiting the system and harming their friends. In very large groups or among strangers there would not be a strong enough social incentive to participate responsibly. This pressure is continually reinforced by the regular face-to-face interaction among the majority of members. In a relatively closed living space, like an apartment or campground, spending behavior reciprocates near-instantaneous feedback. The fact that people are generally aware of each other's spending activity deters selfish behavior. The system also relies on trust in the democratically elected treasurer, who has the power to steal money and deny withdrawals to individual. Over long distances *Kupa* would fail due to the lack of up-close-and-personal social pressures and feasible cash transfer mechanisms. A properly reengineered *Kupa* system must eliminate the need for proximity and centralization while mitigating the risks of trust breaches to allow for a more diverse variety of communities to benefit from its use.

2 Design

CryptoKupa is a totally decentralized system. Each member downloads the CryptoKupa client and the Bitcoin blockchain to their own machine in order to participate. Members only need a record of the blockchain from the time their community implemented Crypto*Kupa*. As a community member, each user has three roles: Contributor, Spender, and Approver. The following protocol outlines the steps that will be followed by each participant's CryptoKupa client in each of the roles.

Variables:

$M = \{A, B, C, D, E\}$

- Set of each member's public key

m-of-n

- Multisig transaction type
- m: Minimum number of signatures needed to release the encumbrance
- n: Number of public keys recorded in the script as possible signers

$L =$ Ordered list of every possible combination of n member

- If $n = 2$

  $L = \{AB, AC, AD, AE, BC, BD, BE, CD, CE, DE\}$
- If $n = 3$

  $L = \{ABC, ABD, ABE, ACD, ACE, ADE, BCD, BCE, BDE, CDE\}$

$c =$ Chunk size (BTC)

$U =$ set of all unspent transaction outputs (UTXO's) in the fund
$U' =$ set of all UTXO's in the fund with at least 6 confirmations
$U'X =$ set of all elements in U' that include member X as a possible signer

Initialization:

In the initialization phase the group decides on the following parameters for their *Kupa* system. They choose a chunk size *c* and an *m*-of-*n* multisig scheme. Then based on *n* and the set of members, *M*, they construct *L*. It is important that each member keep *L* in the same agreed upon order. The consequences of choosing different parameter configurations are explored later.

Example:

M = {A, B, C, D, E}
m-of-*n* = 2-of-2
L = {AB, AC, AD, AE, BC, BD, BE, CD, CE, DE}
c = .05BTC

Contributing:

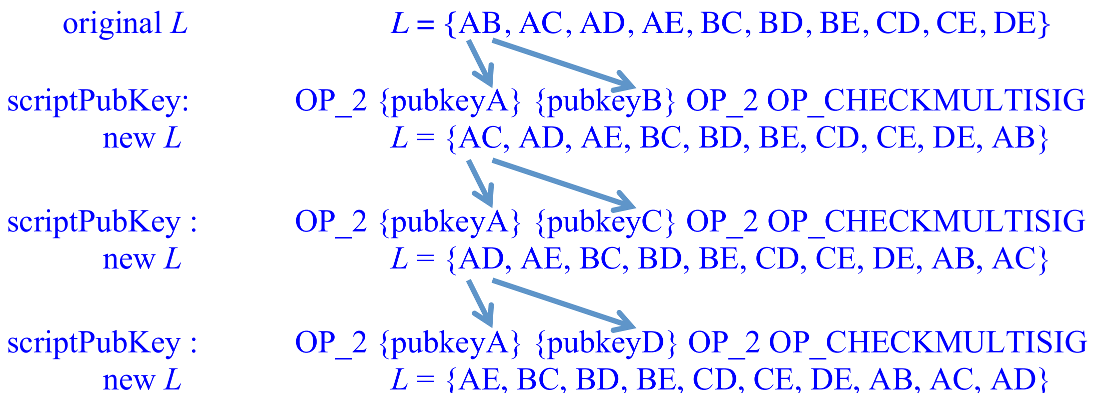
All contributions into the fund must be made in equal chunks of size *c*. For each chunk an *m*-of-*n* multisig is used to lock the transaction to every public key in *L*[0]. *L*[0] is then popped and pushed to the end of *L* and the process is repeated for every chunk.

The list *L* must be kept the same at all times among all members. So when a participant wishes to make a contribution, it must check *U* to see which element of *L* the last contribution into the fund was locked to. Based on this information the new contributor can construct *L* in the appropriate order.

D Contributes .15 BTC:

Repeat the following 3 times:

1) D creates a 2-of-2 multisig tx's of .05BTC locked to *L*[0]
2) Pop/push *L*[0] to the end of *L*



original *L* *L* = {AB, AC, AD, AE, BC, BD, BE, CD, CE, DE}

scriptPubKey: OP_2 {pubkeyA} {pubkeyB} OP_2 OP_CHECKMULTISIG
new *L* *L* = {AC, AD, AE, BC, BD, BE, CD, CE, DE, AB}

scriptPubKey : OP_2 {pubkeyA} {pubkeyC} OP_2 OP_CHECKMULTISIG
new *L* *L* = {AD, AE, BC, BD, BE, CD, CE, DE, AB, AC}

scriptPubKey : OP_2 {pubkeyA} {pubkeyD} OP_2 OP_CHECKMULTISIG
new *L* *L* = {AE, BC, BD, BE, CD, CE, DE, AB, AC, AD}

Spending:
To make a withdrawal from the fund, a member must get the approval of other members, where number of approvals needed is dependent on amount requested. The set of chunks in the system that a member X could potentially spend is represented by *U'X*, as this is every chunk on which X is a cosigner. However, X needs the signature of at least one other member (depending on the type of multisig used) in order to spend any given chunk. So to make a withdrawal of size *w* (BTC), member X would need *w/c* chunks. For each chunk, X must write a transaction transferring the chunk to himself and then request the signatures of all of its cosigners. After they sign and send back the transaction, X signs it as well and broadcasts it to the Bitcoin network. X must only request signatures from cosigners on the chronologically first *w/c* chunks.

A spends 0.1 BTC

1) A constructs an up-to-date *U'A*
   Ex) *U'A* = {AB1, AC1, AD1, AE1, AB2, AC2…}
2) A creates a tx transferring the first 2 elements of *U'A* to his private wallet
3) A tells B and C he wants to spend .1BTC
4) If they approve, B and C sign each transaction and send them back to A
5) A signs the tx's then broadcasts them
   scriptSig1: <sig A, sig B>
   scriptSig2: <sig A, sig C>

Approving:
Members will receive requests for approval when another member tries to spend a chunk on which they are cosigners. When a participant receives an approval request to spend *w* from member X, they first verify that the chunk in question is in fact one of the first *w/c* elements of *U'X*. Then it is up to the approver to decide whether or not she will sign the transaction. This decision is subjective and will be made based on the nature of the spending request in relation to agreed upon community standards and pre-established spending guidelines rules.

C approves A's request

1) C receives A's request to approve chunk *Q* as part of a .1BTC withdrawal
2) C constructs *U'A* and verifies that Q is one of its first two (*w/c*=.1/.05) elements
3) If C approves, then she signs the tx and sends it back to A

Maintaining Trust:
In order to check that everyone is following protocol, each participant's client will monitor transactions that enter and exit *U'*. This will deter dishonesty among participants in their spending and contributing activity. Possible infractions include the following: 1) Putting less than *c* on contribution tx's. 2) Signing contributions to members in any configuration other than what the proper order of *L* specifies. 3) Requesting approval from members for chunks that are out of chronological order in *U'X*. Each of these infractions is easily detectible by a participant's blockchain scanning CryptoKupa client. A member flagged as dishonest will immediately find the communal fund useless to himself, as other members aware of his dishonesty will refuse to approve his withdrawal requests.

3 Design Choices

CryptoKupa's design allows communities to fine-tune its configuration by varying parameters to fit their needs.

m-of-*n* multisig:

- As *n* increases the individual spending cap, MAX, increases
  - MAX: the maximum proportion of the fund that any individual can spend given 100% approval
    $MAX = \frac{|U'X|}{|U|}$
    Assuming $|U'| \gg |L|$, $MAX = \frac{n}{|M|}$
    $^*|U'| \gg |L|$ if the number of contributions made is much higher than the number (members have cycled through *L* multiple times)
  - Ex)
    - If there are 10 members and:
      - If n=2, MAX = 2/10 = 20% of the fund.
      - If n=3, MAX = 3/10 = 30% of the fund
- The relationship between *n* and *m* determines how easy it is to spend any given chunk.
  - If *n=m* then approval is slower and less certain. 100% of the cosigners of a given chunk must sign in order for it to be released. One member's refusal to sign would make spending the chunk impossible.
  - If *n>m* approval is faster and more likely. Only $\frac{m}{n}$ cosigners must sign to release the chunk. The chunk can still be released if less than *n-m*
  - The higher *n* is in relation to *m*, the more vulnerable the fund is to maliciously colluding participants. Given high *n* and low *m* every member is a cosigner on many chunks and it takes small proportion of approvals to release a chunk. The higher proportion of intersection among the same

    members on many chunks gives colluding groups a relatively higher proportion of control.
  - Thus, there is a tradeoff between [approval time] and [MAX + Threat of Collusion]; lower approval time means higher threat of collusion or excessive individual withdrawal

c (chunk size)

- As *c* decreases, spending flexibility/precision increases and total tx fees increase

Approvals-per-BTC = (m-1)/c

- Varied to adjust "difficulty" of withdrawal

4 Conclusion

CryptoKupa represents a way to extend the enactment of community across traditional boundaries of time and space. It allows a group of people to maintain the bond of interdependence that empowers them both socially and functionally without sustained proximity. Any one dishonest member cannot withdraw money or gain any other advantages without peer approval. Furthermore, collusion among subgroups is unlikely to succeed profitably due to the system of chunk-based power distribution. However, while the system is trust-proof (safe without trust), it is not trustless. CryptoKupa can only be beneficial if there is some level of meaningful trust among participants. Because spending power is independent of amount contributed, there is no guarantee that any given member will contribute reasonable amounts. Thus, a genuine trust in the community must exist to prompt participants to give to *Kupa* when they could just as easily keep the money to themselves.

There is extensive potential for real-world implementations of the CryptoKupa protocol. One example is a mobile application that members would use to contribute, approve, and request funds. This would include messaging and picture-sharing functionality that allows users to send receipts for reimbursement and justifications for expenditures. CryptoKupa could be used as a money sharing system for serious endeavors of various sizes or to merely make communal living more communal. Not only does it offer communities a safe and easy virtual sharing infrastructure, but also acts as a tool for strengthening internal relations. It is my hope that CryptoKupa will serve as an inspiration for future research and development of decentralized communities and people-centric structures based in the blockchain.

Acknowledgment

I am grateful to Alex Kuck for his mentorship and advice throughout the development of this work. I would also like to thank Tori Auerhan and Marissa Friedman for helping me to articulate and apply the fundamentals of *Kupa*.

References

[1] http://www.habonimdror.org/about-us/

[2] https://github.com/aantonop/bitcoinbook/blob/develop/ch05.asciidoc#get_utxo_run