Crowbe: An Approach to a Crowd Banking System

Jan C. Beeck
jcbeeck@gmail.com

January 2015

Abstract

A transparent collaborative financial approach through the protocol of cryptocurrency and descentralization of transactions on the Internet. An environment for micro loans based on points where the benefits and risks are shared in a community through prorations based on the weight of the lender. This approach can be used for people who do not participate no have access to the financial institutions.

1 Introduction

Is it possible to create or have a crowd bank system? In recent times there has been a change in attitude towards the use and acceptance of community financing[1,2]. Our approach is an online crowd platform for micro loans. Transparency is a necessary condition, the proposal is conceptualized for sharing the benefits and risks of this venture. In the past, one of the obstacles presented was the lack of a mechanism for virtual currencies that have no third-party connection. Recently there has been a shift in that direction with the appearance of Bitcoins[3], a digital cryptocurrency that is being accepted more and more by people and business in general. Also, Bitcoin has become a native currency on the Internet. Our approach will work with its own "universal currency" known as points; the idea is to exchange the latter to local currencies or to virtual tokens in a rapid and simple way. By using points our approach want to descentralize and create a global platform using the Internet as the backbone.

At the moment half the world does not have access to the financial systems. One example of the use of this approach is for instance a group of people from the first world can help a small entrepreneur in a poor country through micro loans. We believe that money can unite people in such a collaborative environment. Our approach is inspired on Youtube's model where people share videos, why not share money? Sharing means more for all of us!

[1] http://blogs.hbr.org/2013/09/the-end-of-banks-as-we-know-th/
[2] https://www.lendingclub.com/
[3] https://bitcoin.org/bitcoin.pdf

We have many challenges ahead like for instance, the problem of the online wallet service exposed to hackers and the use of the approach by malicious actors. We are also focusing to create a platform on the Web in order to produce a bottom-up innovation for other applications that could be conceived with this crowd banking approach. This approach is aimed to resolve the situation of more than two billion of adults on the planet today without access to banks, credit cards, or to the mainstream financial system.

2 Proposed Framework

Each one of the items described below is a conceptual module to be implemented.

2.1 Transparency

Transparency is the basis of our proposal. We want a system to share money in a collaborative environment. It is not oriented to greed. We propose to have a public log of all transactions involving input, prorating and output of the points.

2.2 Prorate

This is the main concept in our approach, we define prorate as the distribution of an amount proportionately among several users. With the Prorate function we would reduce fraud and use the benefit of sharing profits and risks in a community. The method we propose works based on a weighted array, that means the lender who puts more points into the system is located in the first position, for instance we have this array at the beginning: User = [10, 30, 5, 0, 0], then a new lender puts 50 points into the system, the points are inserted and the array is sorted, after that we have that User = [50, 30, 10, 5, 0]. There are two process related with the prorate: the mining and the insert processes. The first one is when the system borrows points and the second is when the borrower returns the points.

Mining process: This process is called when a borrower requests points, first it is calculated how much points can be borrowed, this is done by dividing each of all the elements of the User array by the number of lenders (i.e. 50/4), after that function we have the available points for borrow per lender = [12.5, 7.5, 2.5, 1.25, 0], the summation of all the elements is 23.75, that is the number of points that can be borrowed. In this example the borrower wants 15 points, because 15 is less than 23.75 the process continue and the system calls to the Prorate function which gives a new array Prorated = [7.5, 4.6, 2, 0, 0], this last one we subtract it with the User array in order to have the new accounts statements which is User = [42.5, 25.3, 8, 5, 0], and the new available points for borrow per lender is [5, 2.83, 0.5, 1.25, 0]

Insert process: This is done when a borrower returns the points, for instance 10 points to the system. The system has the User array, then it calls to the Prorate function and returns Prorated = [5, 3.33, 2, 0, 0] and finally it sums up both arrays and the system has the new accounts statements, in our example the User array finishes with theses values [47.5, 28.66, 10, 5, 0].

In summary the user from the first position had at the beginning 50 points after the first prorated he had 42.5 and finally when a borrower returns 15 points, he finished with 47.5.

2.2.1 Rules

- The system can lend points only if the value requested is less than the sum of all the available points for borrow of all the lenders.
- The "reward" in this approach comes from the last prorate before the account has more points than it had at the beginning. To ensure the profit we have a function that calculates the lower profit acceptable (1), that means the system will prorate while it is less than that value.
- Only can be prorated the accounts that have less points than their Limit.

$$Limit = invested + \left(\frac{invested * 2.9}{100} + 0.30\right) \tag{1}$$

```
#Example of a simple prorate with a weighted array
#Input:
#values = [0, 0, 0, 0, 0, 0, 0]
#pivots = [45, 0, 0, 0, 0, 0, 0]
# 5 is the number of lenders(users) and 45 are the available points

#Prorate function
def Prorate(values,pivots,users,points):
i = 0
j = 1
acum = 0
while i < users:
w =  pivots[i] / (users - i)
x = w * (users - j)
y = x / (users - i)
prorate = w + y
values[i] = prorate
acum = prorate + acum
z = points - acum
pivots[i + 1] = z
i = i + 1
j = j + 1

#Output
values = [16.2, 12.6, 9.0, 5.4, 1.7, 0, 0]
```

2.2.2 Reward

The reward is the "return of investment", instead of using some percentage gain, we define the profit as the extra ammount of points after applying the prorates. Table 1 shows the user who puts 50 points after the first prorated he has 42.5 points when a borrower request 15 points; in table 2 we can see that the same user after 4 prorates ends with 2.5 points of profit, then the user who puts 30 points finished with 2.66 points of profit, and the user who starts with 10 points gains one point after 4 prorates.

Statement	Limit	Γ	Δ	Φ	Γ	Statement
50	51.75	12.5	15	1	5	42.5
30	31.17	7.5	15	1	2.83	25.3
10	10.59	2.5	15	1	0.5	8
5	5.44	1.25	15	1	1	5

Table 1: Table of the mining process based on an initial investment, Limit represents the minumun profit, Γ are the available points to borrow, Δ are the borrowed points and Φ are the number of prorates.

Statement	Ψ	Statement	Ψ	Statement	Ψ	Statement	Φ	Profit
42.5	10	47.5	5	50	5	52.5	4	2.5
25.3	10	28.66	5	30.66	5	32.66	4	2.66
8	10	10	5	11	5	11	4	1
5	10	5	5	5	5	5	4	0

Table 2: Table of reward, Ψ are the points returned to the system and Φ are the numbers of prorates applied in order to have a profit for a lender.

2.3 Cryptography

Cryptography is the study of mathematical techniques related to information security. It aims to ensure confidentiality, integrity, authentication and irreversibility of the information [1]. The asymmetric cryptography, also known as public key cryptography, uses a pair of keys, one of the keys is used to encrypt and the other for decoding. These pair of keys consist of a public key known to all and a private key, the latter must not be disclosed to third parties under any circumstances. Usually the public key is used to encrypt the message and the private for decryption. Asymmetric systems use mathematical functions in their algorithms.

At the core of cryptography-based transactions is the descentralized consensus, a scheme that transfers authority and trust to descentralized network and enables its

nodes to continuously and sequentially record their transactions on a public 'block', creating a unique 'chain', the blockchain which is its backbone [4].

2.3.1 Cryptocurrency

Electronic cash is one of the most important application of modern cryptology know as cryptocurrencies, they use a public key and a private key (asymmetric cryptography systems), the latter known only to the owner of the currency. An ideal cryptocurrency according to [2] has the following properties:

- Independence: The security of electronic cash cannot depend on any physical condition. Then the cash can be transfered through networks.
- Security: The ability to copy (reuse) and forge the cash must be prevented.
- Privacy: The privacy of the user should be protected; the relationship between the user and his purchases must be untraceable by anyone.
- Off-line payment: the transactions can be done without requiring a third party to validate the transaction between the two involved parties (buyer and seller).
- Transferability: The cash can be transfered to other users.
- Divisibility: One value X of cash can be subdivided into many pieces such that the cash subdivided in pieces is worth any desired value less than X and the total value of all pieces is equivalent to X.

2.3.2 Cryptocurrency Wallets

The wallets or clients are software used to implement various cryptographic protocols currencies. They provide communication services to the network, perform transactions, data can be saved locally and perform verification and disclosure of transactions. One of the main functions is to keep transactions and private keys associated with the cryptocurrency. Figure 1 shows the strong connection between Crowbe and the cryptocurrency protocol as well as the integration with a wallet (i.e Bitcoin based-wallet [5]).

3 Discussion

- The fee: Crowbe could charge one point of all the available points to borrow every time the system calls the function Prorate (i.e the fee might be 0.0001 BTC).

[4] http://techcrunch.com/2015/01/18/after-the-social-web-here-comes-the-trust-web/
[5] https://blockchain.info/api

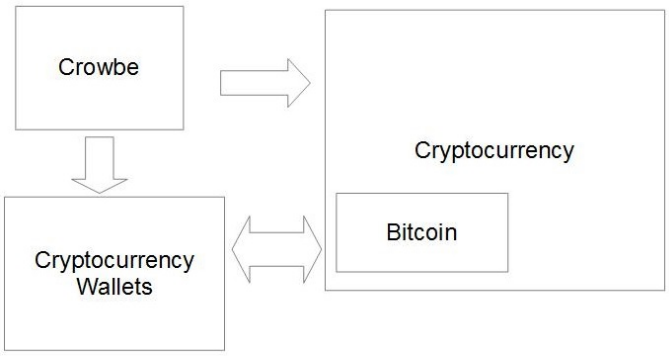

Figure 1: Crowbe's interaction with the cryptocurrency programmability.

- Security fund: apart from sharing the risks throw the Prorate function it also could be used some security fund, that means from all the available points to borrow, the system could keep 16% and lend the rest.
- Interoperability: Crowbe could be used as a crowd-bank-engine.

References

[1] Menezes A.J.; Van Oorschot P.C.; Vanstone S.A. *Handbook of Applied Cryptography.* Waterloo: CRC Press; 1996.

[2] Okamoto T.; Ohta K. *Universal Electronic Cash.* In: Springer. Advances in Cryptology - CRYPTO'91; 1991 August 11-15; Santa Barbara, California. Berlin: Springer; 1992. p.323-337.

[3] Moia V. H. G.; Henriques, M.A.A. *Avaliação da Segurança de Protocolos Criptográficos Usados em Moedas Virtuais.* In: VII Sétimo Encontro dos Alunos e Docentes do Departamento de Engenharia de Computação e Automação Industrial - EADCA,Vol. 1, pp.1-4, Campinas, SP, Brasil, 2014.