# Implementation of Digital Cash

Project Report†

Rathna Ramesh
San Jose State University
Computer Engineering Department
rathna.ramesh@sjsu.edu

Sphoorti Metri
San Jose State University
Computer Engineering Department
sphoorti.metri@sjsu.edu

Abhilash Garimella
San Jose State University
Computer Engineering department
abhilash.garimella@sjsu.edu

## ABSTRACT

Digital Cash is a system that adopts properties from both the real world cash and the e-cash we pay online. It is built around the anonymity and transferability property of the real world cash as well as the ease of usage associated with the e-cash. The transactions of the digital cash are neither recorded nor associated with a particular spender/ customer which overcomes the major drawback of the e-cash system – lack of anonymity.

## CONCEPTS

• **Network Security:** RSA, HMAC, Blind Signatures, Secret Splitting, Commitment Protocol, Double Spending

• **Programming Language:** Python

• **Computer Networks:** Socket Programming

• **Tools:** Jupyter Notebook

## KEYWORDS

Digital Cash, Blind Signatures, Secret Splitting, Commitment Protocol, Double Spending

## 1 INTRODUCTION

Since the inception of PayPal in 1998, the usage of electronic payment methods has always been on the rise. This cash need not be stored or secured (which is taken care by the bank) and is easily payable which is a major advancement when compared to the real world physical cash. The disadvantage of using this system is that it is not anonymous (a click on your payment history in your bank records could show you all the transactions you have done since having an account in the bank) and the lack of offline transfer properties (The EVM/ Mobile needs to be online all the time) which are not desirable. Hence, we needed a system that could have the best of both worlds. The major inspiration behind the digital cash system came when Satoshi Nakamoto published his paper about Bitcoins on 31 October 2008 where he introduced the idea of Peer-to-Peer Electronic cash system. **Fig. 1** shows the architecture of the implementation of the digital cash system. The Bank oversees all the transactions but cannot associate a transaction to a particular sender-receiver pair. This is achieved using Blinding protocol. Bank is also involved in the verification of each payment and prevention against double spending of the cash.

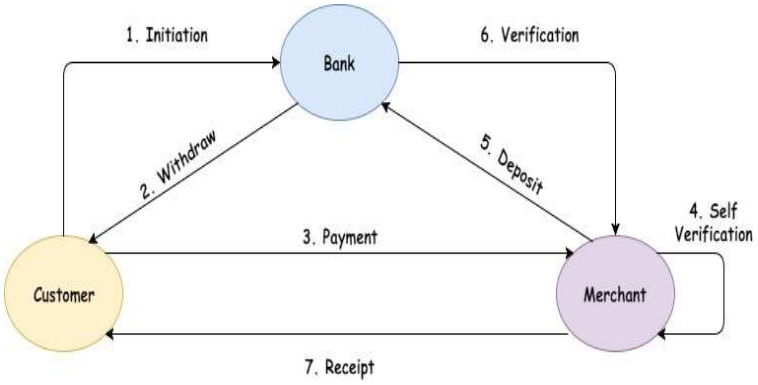


**Figure 1: Flow of the cash in the Digital Cash**

**System.**

The Merchant is responsible for the verification against the correct amount sent to him, reporting to the bank in order to withdraw the money and sending the receipt to the customer. This process could be done online or offline as digital cash has the properties of being stored or copied in devices like a memory stick or a flash drive.

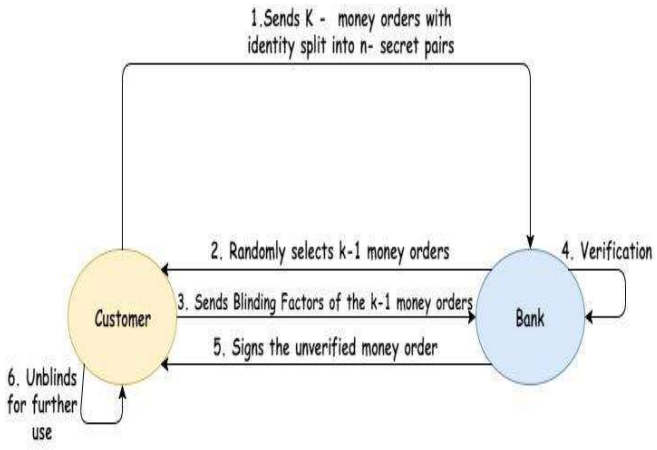


**Figure 2: Blinding Protocol and secret splitting**

**Advantages of Digital Cash:**

a. The cash spent is not associated with a particular user. This protects the anonymity of the spender and makes the cash untraceable.

b. The verification process at the bank makes sure that the cash is not double spent by the customer or merchant.

c. The hardware requirements for such an implementation is very low.

d. Deployment of the processes on a larger scale is easy.

**Disadvantages of Digital Cash:**

a. There is a need for extra communication between the merchant and the bank in order to verify that the money has not been double spent.

b. The cash strings are not usable once encased in the bank. New string need to be generated all the time. Hence the amount of resources that need to allocated is high.

c. There is problem with scalability of the model. Special software need to be run for the synchronization of all the process across the bank as well as the merchant.

d. The databases that need to be maintained by the bank are to be huge. The strings literals keep on changing and hence the rate of data growth is also very high.

## 2 BLIND SIGNATURE PROTOCOL

This protocol is implemented in order to achieve **anonymity** property. For this purpose, it makes use of a blinding factor.

a. Customer sends **r = (m* (b^e) mod n)** to bank. Here, e is public key and b is the blinding factor. It sends k such money orders with identity split into n secret pairs (**Secret Splitting**). Bank selects k-1 orders and asks the customer for their corresponding blinding factors.

b. Customer replies with the blinding factors of the k-1 money orders requested by the bank.

c. Bank verifies all the k-1 money orders, checks if the customer has put the same amount in all

of the orders. The probability of the customer cheating with the unverified money order decreases with the increase in the value of k. The bank signs the unverified money order if the customer is not trying to cheat. If the customer cheats, the process will be terminated.

d. Bank signs the unverified money order with its private key (d) and sends it back to the customer.

**r^d = ((m*b^e) ^d mod n) = ((m^d) * (b) mod n)**

e. Customer can now remove the blinding factor for further use.

**(b^-1)[r^d] = (m ^ d) mod n**

In the above equation, the customer applied the inverse of the blinding factor that was used to hide the message from the bank. He now has the signed copy of the money order he requested for.

f. The customer now can store the money order, spend it somewhere or copy and send the string to anyone. Fig 2 explains the working of the blinding protocol

## 3 SECRET SPLITTING

Secret splitting is used to split a single message into n parts. Each part is meaningless by itself and makes sense only when all the parts are combined together.

- A Random Number Generator is used to generate keys.
- Message is XORed with the key
- The message is now split into key and the result (message XOR key = result)
- Only when the key and result are XORed together, message can be obtained.

An identity string belongs to the owner of the coin. This identity string is split into parts using secret splitting. This procedure helps in detecting double spending. Initially the identity of the user is concealed but can be tracked in case of double spending.

When a customer spends the money, one among each of these pairs are sent to the merchant, ensuring that no complete pair is present. This way, the owner of the money is not identified.

If a customer makes another copy of the money order and tries to use it for the second time, the issuer can track the culprit. This is achieved by combining corresponding identity pairs of the original and duplicate money orders.

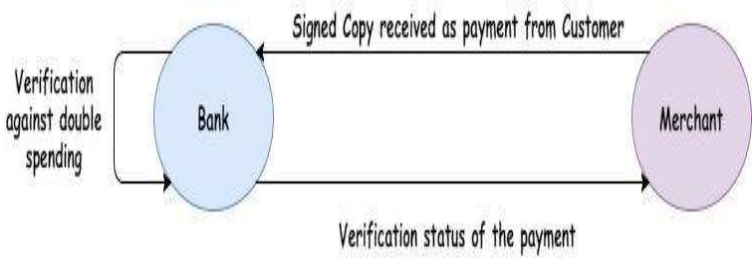


Fig 3: Verification against double spending

## 4 COMMITMENT PROTOCOL

The following is the commitment protocol that takes place between customer and merchant:

a. Customer has the digital cash but wants to keep it a secret from the merchant for now and would like to reveal it to the merchant after some time.
b. The merchant wants to make sure that, customer has revealed the same digital cash that he promised to earlier.

Implementation makes use of one way and collision free property of hash functions.

a. Customer and Merchant agree upon a hash function.
b. Customer applies HMAC on the message using a secret key and the hash function
c. He sends only the key and hash output to the merchant.
d. The merchant stores these values for later verification.
e. Customer later sends the message to merchant for verification
f. Merchant now applies HMAC to the message that has been sent by the customer.
g. If the hash value obtained and the one stored before match, the merchant proceeds with the transaction by sending the money order to the bank.
h. If the hash values do not match, the merchant terminates the process as part of fraud detection.

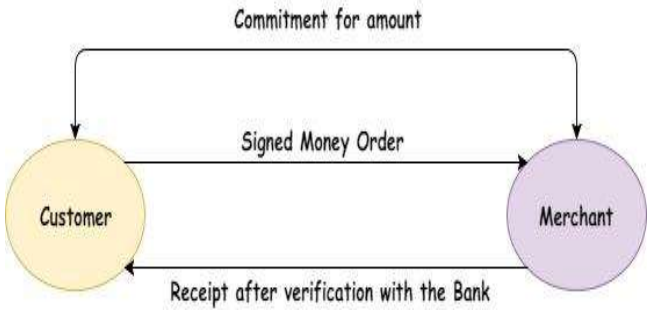


Fig 4: Commitment process

## 5 RESULTS AND DISCUSSIONS

The entire process can either complete successfully or terminate in one of the following ways:

a. Termination due lack of funds: Unavailability of funds in the account of the customer would lead to the termination of process unless the customer deposits more money into his bank account.
b. Termination due to double spending: Bank has the ability to detect double spending by comparing the split identity pairs of the owner of the money order.
c. Termination due to false commitment of the customer to the merchant: If the hash values that have been sent to the merchant do not match, the process terminates and the customer receives a message from the merchant regarding the transaction.

## 6 CONCLUSIONS

In summary, we have implemented the digital cash system in such a way that it offers online as well as offline modes of transfer of money while providing features like anonymity, prevention against double spending and fraud detection. We have used Commitment, Secret Splitting and Blind Signature protocols to implement the system. The program is user friendly in nature and can be executed easily.

The offline digital cash system is similar to the one we implemented. The difference lies in the customer using secure alternate means to transfer it to the merchant.

## 7 Python Libraries:

a. RSA
b. PyCrypto
c. HMAC
d. random
e. BitVector
f. Numpy

## 8 References

[1]http://pages.cs.wisc.edu/~jha/course-archive/642-fall-2010/project/Cash.html
[2]http://www.cs.mcgill.ca/~crepeau/CRYPTO/BCDemo/BCprotocol.html
[3]https://pdfs.semanticscholar.org/9c6a/3ff9fe9895109779ff38972d9edd1422f460.pdf