Digital Currency Generation Using Hash Function and Digital Signature (DigCur)

Qusay Mohammed Jafar
Department of Computer Communication Engineering, Al-Rafidain University College, Baghdad-Iraq.
Corresponding author: qusay.mj@ruc.edu.iq.

Abstract
This paper proposed a strategy to create a digital currency by producing a list of serial numbers by an authorized organization (Issuer). Each number in the serial numbers list will be merged with the issuer’s secret key (Sec_key) and then hashed using hash function to produce the issuer hash code (H-code). The issuer hash code will be considered as a digital banknote and then it will be distributed to the payee as a type money. When a person likes to pay (transfer) these digital banknotes to another person, he/she must prove this transaction by his/her privet key using digital signature process. This mechanism needs a third party to control the process of money transaction by checking the validity of the digital signature and who is authorized to own this digital banknote. The third party is the issuer himself (the central bank). For each digital banknote the issuer hash code will be saved in a record without changing to check if this digital banknote is counterfeit or not. The problem of double spending will be solved by swapping the owner identifier immediately after the transaction process. The proposed method aims to produce a digital cryptocurrency to be as alternative to the traditional physical currencies and it will be called as DigCur. According to some tests the proposed method can be considered a good method to take advantage of it instead of other cash applications. [DOI: 10.22401/ANJS.22.2.09]

Keywords: Digital Currency, Secret Key, Hash Function, Digital Signature, Digital Banknotes.

1- Introduction
Digital currency is a type of money but it is formed digitally, also it is a payment method represented in digital form [3]. The digital currency has become a reality, and the modern world requires faster and safer financial transactions, it is therefore necessary to deal with mechanisms which make money transaction more efficient, and this paper proposed a method for generating digital currency with some principles of cryptography.

There are many ways to transfer money from sender to the receiver and all of them depend on the amount of money that will transfer but not the banknote itself, for example; in any electronic payment system to transfer 200$ from the sender’s wallet to the receiver’s wallet the process required a subtraction and addition operations with identification and authentication processes, so all electronic financial transactions not take the money units (banknotes) in the transaction process. The method suggested in this research deal with the banknotes as digital currency and will be called as DigCur.

2- Digital Currency
There are two terms; virtual currency and digital currency, virtual is anything created from nothing, while digital is a process of generating the currency by the computer, thus the term digital is preferred over virtual. In short, the two terms refer to the alternative method of conventional currency [3]. Virtual currencies issued by private agents and there is no control by the governments [15], therefore DigCur is a type of digital currency. In this research the digital currency that will be generated depends on the serial number which will be determined by the central authority (Central Bank).

There are many cryptocurrencies and bitcoin is one of them. Bitcoin depends on special methodology in which; the electronic cash for online payments send the money directly from one point to another without going through the third party [8], also the bitcoin paper proposed a strategy for solving the problem of double spending. Bitcoin is good and new method for cryptocurrency but it is not safe enough to be replacement for real money [10].
Centralized payment systems solve the problem of double spending but it needs some trust by the customers [2], for this reason we need the third party to control the works.

There are some differences between cryptocurrency and digital currency. Cryptocurrency is digital currency in its essence but it is differ in the Structure, Anonymity, Transparency and Transaction manipulation. The digital currency is centralized while cryptocurrency is decentralized, identification is required in digital currency, and in cryptocurrency identification is not required. The process of money transaction in digital currency is not transparent it gives some privacy for the customer wallet, in cryptocurrency the transaction can be seen by anyone because of the public chain. In the digital currency there is a full control by the third party, in cryptocurrency there is no control on the transaction [13].

One of the first attempts of electronic cash is explained in [6] where the researchers applied the digital signature to verify the coins, the system works in offline and online with a centralized database to solve the problem of double spending, the system can detect the double spend in online manner only, but in [6] there is no clear strategy for digital currency generation.

3- The purpose of generating DigCur
Everything in this life is in developing, banking and finance is one of the most applications that must keep up this development. This paper aims to generate a digital currency using the principles of the cryptography. The motives for proposing a new strategy (DigCur) for generating digital currency which depends on hash code and digital signature are:-

1- Safety of currency fraud (no chance for counterfeit money).
2- There is no dual digital banknote (uniqueness).
3- Digital signature prevents the sender from denying the process of money transaction that accomplished by him.
4- Checking the sender and receiver identities (identification and authentication).
5- Control the financial transfer.
6- The third party prevents the double spending.
7- Ease of dealing with banking operations.
8- No need for cash money.
9- New strategy for financial operations.

4- DigCur Mechanism
DigCur is digital currency because it is generated by computer program, and it is Cryptocurrency because it is needs to produce a hash code and digital signature. Sections in below explains how the DigCur will be generated and how the persons can transfer DigCur to others.

4-1 DigCur generation by the central bank
Central bank is the only authorized committee that must be able to generate DigCur, the process of DigCur generation will begin by selecting an initial value to produce a list of serial numbers, each serial number (SN) will be converted to a string and then it will be merged with the secret key (Sec_key) that has been selected by the central bank. The hash function will proceed on the merged string to generate the hash code, H_code=HF("SN"+"Sec_key"), where HF is the hash function, SN is the serial number and Sec_key is the secret key chosen by the issuer, in this research MD5 is the method that has been selected to generate the hash code because of its simplicity. MD5 is faster than SHA [14]. MD5 is described in [11]. In Microsoft Visual basic.net 2013 there is a built in function make easy to compute the MD5 of any input, we used it to compute the hash code.

At this stage the H_code is the core. This manner will simplify the operation of checking whether this currency is counterfeit or not, the process of detecting currency fraud is usually done by a third party (the central bank) which owns the secret key (Sec_key), where the central bank is the issuer himself.

To explain how the DigCur will be generated:- the issuer must select a start point and end point to generate a list of serial numbers and then each one of the serial numbers will be manipulated to generate the DigCur.
For example let;
SN="44670"
Sec_key="hello"
MergeSN_Sec_key ="44670hello"
Then;
H_code=MD5(MergeSN_Sec_key)
H_code = 76da990ed294983ef68c794243afbb50
8 → hash code of 32 Hexadecimal digits, this hash code is unique. And so on for the next serial number (SN).

Table (1) shows a list of digital banknotes that has been generated by the issuer program.

Table (1)  
Some digital banknotes of class 5DigCur with secret key ("hello").

| SN     | Date of Issue | Currency class | Issuer | H_code                      |
|--------|---------------|----------------|--------|-----------------------------|
| 44670  | 04/06/2018    | 5DigCur        | Z_BANK | 76da990ed294983ef68c794243afbb50 |
| 44671  | 04/06/2018    | 5DigCur        | Z_BANK | aefed49a7eb272d80a5c330045ef455  |
| 44672  | 04/06/2018    | 5DigCur        | Z_BANK | 50004773951ec981b0377f19595b2b28 |
| 44673  | 04/06/2018    | 5DigCur        | Z_BANK | e748199d8e786d6e452aad1de40520 |
| 44674  | 04/06/2018    | 5DigCur        | Z_BANK | 92318e9299d206e15b068ea264805b0b |
| 44675  | 04/06/2018    | 5DigCur        | Z_BANK | 682e44d0d0a76f35dfdb78c9d2d2b05b |
| 44676  | 04/06/2018    | 5DigCur        | Z_BANK | e5a3345f75740e6ea51a9452ade43e23 |
| 44677  | 04/06/2018    | 5DigCur        | Z_BANK | 30159aa4c6506028adbc926600ca48 |
| 44678  | 04/06/2018    | 5DigCur        | Z_BANK | 17a73eb239a5e39e9203c3b8e60d8 |
| 44679  | 04/06/2018    | 5DigCur        | Z_BANK | d3a5d13b91404e865a1541d3e1d7f0f |

4.2 DigCur distribution and transaction

After generating the digital banknotes of the DigCur, the list of these digital banknotes will be saved and then will be ready to the process of distribution by the central bank to the payees after signing each digital banknote. In this research work the Elgamal digital signature is the method that has been selected to sign the DigCur, therefor each template of the DigCur will contain the digital signature of the sender and other required data as shown in Table (2). The Elgamal digital signature algorithm is explained in [9][4][7]. In each transaction the process of digital signature will be done by the sender to validate the process of the money transaction. The digital signature will prevent the sender to deny the process of money transfer. Also the DigCur template must have the owner identifier (Owner ID) and this field is dynamic (exchanged with each DigCur transaction).

Table (2)  
DigCur template.

| Serial number | Date of Issue | Currency Class | Issuer | H_code | Digital signature | Owner ID |
|---------------|---------------|----------------|--------|--------|------------------|----------|

4.3 Digital Signature and Hash Code.

Digital signature is the process of signing a message with sender’s private key. RSA, Elgamal are some examples of digital signature methods that originally are public key encryption methods [9][7]. Digital signature is a process to protect the data from forgery and to prove the integrity of the data. Digital signature is a form of cryptographic transaction that makes the receiver to know the source of the message [12].

Hash function is an algorithm that can summarize the message of variable length to a unique and fixed size of n bits, MD5 will produce a code of 128 bits, SHA1 will produce a code of 160 bits [5]. Hash function is one way encryption, for a message x, and it is impossible to find x by reversing the parameters [12], and this will protect the secret
key \((Sec\_key)\) that must be known for the third party only.

Hash function is used in digital signatures and in data integrity [5][1], to simplify the process of digital signature the message must be hashed to produce the hash code, and this because it is useless to sign each character in the message, therefore no need to signing the entire message, but only the hash code of the entire message will be signed, and then the hash code with the parameters of the digital signature will be sent to the receiver, and the validity checking will be done by the receiver, but in this research work the digital signature validation will be done by the central bank (issuer) who is considered as a third party.

MD5, SHA-1, SHA-2 are some examples of message hashing functions [9][1]. MD5 and SHA-0 both of which have been broken [12], therefore we advise to use another hash function such as SHA3.

To compute the hash_code it must to apply the following formula:

\[
\text{hash\_code} = HF(x)
\]

As shown in section 4-2 the sender must signs each digital banknote, the message that must be signed is the \(H\_code\) itself. The role of the hash function in this research is to verify the integrity of the digital banknotes (DigCur) and to apply the digital signature on the hash code of the message instead of an entire message, where the message \((x)\) is the SN and the \(Sec\_key\).

4-4 Solving the problem of double spending

Double spending is the problem of using the same digital currency more than once [16]. In this paper, to solve the problem of double spending; the unique identifier (ID) of the person that will receive the digital banknote will be inserted in DigCur template instead of the sender’s ID, for this the central bank will be responsible for matching the saved template (DigCur record) with the current transferred DigCur to decide if this person is authorized for transaction or not. To make the DigCur transaction more realistic we need a central database that will save copies of the DigCurs for all persons. In the central database there is a hierarchy for indexing the DigCurs based on the serial number to make the DigCur record is easy for retrieving. For each DigCur record and after finishing the transaction process the sender’s ID will be swapped immediately by the receiver’s ID, and this will deny the sender to send the DigCur of a specific serial number again because of the matching and authorization process will be done by the central bank. Fig.(1) shows the mechanism of the DigCur transferring.

5- The proposed DigCur Algorithm

Sections below show the steps required for each stage in the DigCur.

5-1 DigCur generation algorithm

Steps below show the algorithm of generating the DigCur by the central bank (issuer).

Algorithm (1): DigCur Generation Algorithm

Input: Start_SN, End_SN, Sec_Key, DigCur class.

Output: DigCur records.

Step 1: Begin
Step 2: Set the range of the serial numbers between (Start_SN) and (End_SN).
Step 3: Select the secret key (Sec_Key).
Step 4: Set the DigCur class.
Step 5: For each number (SN) in the range between (Start_SN) and (End_SN) must be converted to a string (st).
Step 6: Merge (st) with (Sec_Key) to create the message (M).
Step 7: Compute the hash code (H_code) by applying the hash function (HF) on the message (M).
Step 8: Insert the (H_code) in the DigCur record (template).
Step 9: For the next (SN) go to step 5.
Step 10: Save the DigCur records, where each record must contains some necessary fields (SN, Date of Issuing, Currency Class, Issuer, H_code, empty cells for the next digital signature, and Owner ID).
Step 11: End DigCur generation algorithm.

5-2 DigCur transaction algorithm

Steps below show the algorithm of DigCur transaction operation between the sender and the receiver.
Algorithm (2): DigCur Transaction Algorithm.

Input: SenPrivate_Key, SenPub_Key, DigCur class, Number of DigCur.
Output: Accept or reject the DigCur.

Step1: Begin
Step2: Set the private key (SenPrivate_Key) and the public key (SenPub_Key) by the sender for digital signature operation.
Step3: Set the class and the number of the DigCur by the sender.
Step4: Prepare the DigCur for transmission.
Step5: Apply the digital signature (DS) on (H_code) using (SenPrivate_Key) to compute the (V).
Step6: Send the DigCur to the receiver (as a record) with (V).
Step7: The third party will merge (SN) with the (Sec_Key) and then compute the hash function (HF).
Step8: If (HF) is equal to (H_code) then the DigCur is valid, otherwise reject the DigCur transaction (DigCur is forged) and go to step 14.
Step9: Checking the digital signature validation of the sender by computing the (Value1 and Value2) using (SenPub_Key).
Step10: If (Value1) is equal to (Value2) then the digital signature is valid and the transaction must be accomplished, otherwise reject the DigCur and go to step 14.
Step11: If the sender’s ID match with the copy of the DigCur in the central database then immediately updates the DigCur record by assigning the receiver’s ID in the owner ID cell, otherwise reject the transaction (unauthorized sender).
Step12: Update the sender and receiver wallets by the third party.
Step13: For the next DigCur go to step 4.
Step14: End DigCur transaction.

6- Implementation Results

The proposed algorithm (DigCur algorithm) was implemented using Microsoft VB.Net2013, sections bellow show the results of tracing the system.

6-1 Generating DigCur

As shown is sections above, the central bank (CB) is the only authorized foundation that able to issue the DigCur, suppose the CB likes to issue 20 digital banknotes of class 50 DigCur with a secret key “system2016YaQu”, Table (3) shows the record of each digital banknote and then each record will be transacted to the payee. In the stage of DigCur generating there is no need to the digital signature, digital signature will be necessary in the transaction step.

Table (4) shows the DigCur with the secret key “system2016YaQ3”, you notice there are
new list of hash code differ totally from Table (3), and this for because of hash function specification (MD5 in this case).

After signing each digital banknote, the central bank is ready to distribute the DigCur(s) to the payees.

**Table (3)**

List of DigCur of a proposed class (50DigCur) with secret key “system2016YaQusay Mohammed Jafar

| SN     | Date of Issue | Currency class | Issuer | H_code                | Owner ID |
|--------|---------------|----------------|--------|-----------------------|----------|
| 432431 | 14/4/2018     | 50DigCur       | ICB    | 89bf75b94e3f9bd713a64efb74fd165 | ICB      |
| 432432 | 14/4/2018     | 50DigCur       | ICB    | 6b36d13b6fcb86d924f073a54aace056 | ICB      |
| 432433 | 14/4/2018     | 50DigCur       | ICB    | af9ce04bed31cedbdf1f13d0e4b9f46b | ICB      |
| 432434 | 14/4/2018     | 50DigCur       | ICB    | e03ec1d8806f6af68c60f1eebce3589ab | ICB      |
| 432435 | 14/4/2018     | 50DigCur       | ICB    | e2d4575e06ee83ee96c3e4a72d203406 | ICB      |
| 432436 | 14/4/2018     | 50DigCur       | ICB    | f19e24b78376b6d66707fa7d9940c13 | ICB      |
| 432437 | 14/4/2018     | 50DigCur       | ICB    | 9f928bd5c72a84241fb695cc47b95af | ICB      |
| 432438 | 14/4/2018     | 50DigCur       | ICB    | 5af57605f5016df656898ce458abf0 | ICB      |
| 432439 | 14/4/2018     | 50DigCur       | ICB    | fb7e272949ac2d633ca097e76e6f0a47 | ICB      |
| 432440 | 14/4/2018     | 50DigCur       | ICB    | e2507555a22e380db8073e039ab9d884 | ICB      |
| 432441 | 14/4/2018     | 50DigCur       | ICB    | bdbf7dbd355c851a23ce9d733878a868 | ICB      |
| 432442 | 14/4/2018     | 50DigCur       | ICB    | 228d3b5e760f4ec38310807c3bca12 | ICB      |
| 432443 | 14/4/2018     | 50DigCur       | ICB    | 47990c8596a743d54b04b13d0e2eb7 | ICB      |
| 432444 | 14/4/2018     | 50DigCur       | ICB    | deaafacb739f9204287fd4d44f6b4 | ICB      |
| 432445 | 14/4/2018     | 50DigCur       | ICB    | 1ae19b991e0c8e63e9470540976d1c | ICB      |
| 432446 | 14/4/2018     | 50DigCur       | ICB    | b2facerf37643a992d6b8128f70a4f | ICB      |
| 432447 | 14/4/2018     | 50DigCur       | ICB    | 6d25992bf21426b9207de901387e4c | ICB      |
| 432448 | 14/4/2018     | 50DigCur       | ICB    | 4acdeb3a723a584cd4159f61a3fd6c5 | ICB      |
| 432449 | 14/4/2018     | 50DigCur       | ICB    | 6a18e867043dbf209a0313e76684120 | ICB      |
| 432450 | 14/4/2018     | 50DigCur       | ICB    | 4b5e81ab1f0d1135d04ad85f1a91d4 | ICB      |

**Table (4)**

New list of DigCur of a proposed class (50DigCur) with secret key “system2016YaQusay Mohammed Jafar

| SN     | Date of Issue | Currency class | Issuer | H_code                | Owner ID |
|--------|---------------|----------------|--------|-----------------------|----------|
| 432431 | 14/4/2018     | 50DigCur       | ICB    | c86469d26e9a89b3e0a123d1a1091f4 | ICB      |
| 432432 | 14/4/2018     | 50DigCur       | ICB    | 710200e379d9909fde467b41a0f5c97 | ICB      |
| 432433 | 14/4/2018     | 50DigCur       | ICB    | b7470aeb19f529820ec3df1f6293899f | ICB      |
| 432434 | 14/4/2018     | 50DigCur       | ICB    | c2eb69005ed89111e22cf9bba8053f90 | ICB      |
| 432435 | 14/4/2018     | 50DigCur       | ICB    | c3b0b8e21e77797c9e3c9eb0f92898d8c | ICB      |
| 432436 | 14/4/2018     | 50DigCur       | ICB    | 87eb8e71a1956d070e106e4c6f6d26f | ICB      |
| 432437 | 14/4/2018     | 50DigCur       | ICB    | d4717029f87c90b989e9e0f8961a50f | ICB      |
| 432438 | 14/4/2018     | 50DigCur       | ICB    | 0888f559b781af9b36a831e1c4f860248 | ICB      |
| 432439 | 14/4/2018     | 50DigCur       | ICB    | d25f6735049f43def88848696ee166d7c | ICB      |
| 432440 | 14/4/2018     | 50DigCur       | ICB    | 46d87ac58a755e11f15f50c60991e4c | ICB      |
| 432441 | 14/4/2018     | 50DigCur       | ICB    | 03b168b068a978b404f12124b665c | ICB      |
| 432442 | 14/4/2018     | 50DigCur       | ICB    | 9e23a5552ba28968752d3e2967b7ae | ICB      |
| 432443 | 14/4/2018     | 50DigCur       | ICB    | a58be177025615b9929e6a77245b8c | ICB      |
| 432444 | 14/4/2018     | 50DigCur       | ICB    | a293b34a576df670eb50e779f9954d06 | ICB      |
| 432445 | 14/4/2018     | 50DigCur       | ICB    | d92d340a607a9c1da206c17f62776 | ICB      |
| 432446 | 14/4/2018     | 50DigCur       | ICB    | 8c1307450527c7f50717a5356019e8 | ICB      |
| 432447 | 14/4/2018     | 50DigCur       | ICB    | 48d282d92465b527a16a3b33b87d08d9 | ICB      |
| 432448 | 14/4/2018     | 50DigCur       | ICB    | 362ddcc5e1085976f1a2bb50eab327 | ICB      |
| 432449 | 14/4/2018     | 50DigCur       | ICB    | ad3f8d8069c2761ca0998b492505e19 | ICB      |
| 432450 | 14/4/2018     | 50DigCur       | ICB    | 0e3d85a201e385f663abcb6c493f187 | ICB      |
6-2 DigCur transaction

To explain how the DigCur will transfer from one person to another, suppose the person (Person1) owns the following digital banknote and he wants to transmit it to (Person 2).

| SN     | 432432 |
|--------|--------|
| Date of Issue | 14/4/2018 |
| Currency class | 50DigCur |
| Issuer | ICB |
| H_code | 710200e37d9909fdbc467bf41a0fc397 |
| Owner ID | Person1 |

To explain how the DigCur will transfer from one person to another, suppose the person (Person1) owns the following digital banknote and he wants to transmit it to (Person 2).

First the sender must sign each digital banknote and then the third party will complete the mechanism of the money transaction.

In this paper we select Elgamal digital signature method because of its simplicity.

To apply the Elgamal digital signature there are some computations (in below a tracing of the Elgamal Digital signature):

In the sender side:
- Let \( p=1753 \) (\( p \) is primary number, to increase the complexity; \( p \) must be large prime number)
- \( a=8 \) (\( a \) is primitive element to \( \mathbb{Z}_p \))
- \( d=12 \) (\( d \) must be between (2) an (p-2)), it is the sender’s private key.

Compute:
- \( B=(a^d \mod p)=1026 \)
- \( (P, a, \text{ and } B) \) is Person1’s public key, where Person1 is the sender.

Send \( p, a, \text{ and } B \) as public key to the receiver to verify the digital signature.

```
P=1753  a=8  B=1026
```

Now it must to compute the digital signature of the message, the message (x) is the hash code of the digital banknote.

For the digital banknote in the above;
\[ x = 710200 \text{ e37d99 09fdbc 467bf4 1a0fc3 97 } \]
and to simplify the computations and to overcome the problem of very large integer numbers with exponential; we partitioned the message (x) to six sub-messages (\( x_1, x_2, x_3, x_4, x_5, \text{ and } x_6 \)), first five sub-messages of size 24 bits, the last sub-message of size 8 bits, the total number of bits is 128 bits (MD5 hash code size).

\[
\begin{align*}
  x_1 &= (710200)_{10} = (7406080)_{10} \\
  x_2 &= (e37d99)_{10} = (14908825)_{10} \\
  x_3 &= (09fdbc)_{10} = (654780)_{10} \\
  x_4 &= (467bf4)_{10} = (4619252)_{10} \\
  x_5 &= (1a0fc3)_{10} = (1707971)_{10} \\
  x_6 &= (97)_{10} = (151)_{10}
\end{align*}
\]

The third party will apply the two steps of the verification:

First; it will check the hash code if it is accepted or not, by inserting the central bank secret key (\( \text{Sec_key} \) “system2016YaQ3” with the SN “432432” and then compute the hash code, if it is accepted, then the second process will begin by checking the digital signature using the sender’s public key (1753  8  1026).

Now; Let \( \text{Ke}=7 \), where \( \gcd(\text{Ke}, p-1)=1 \)

For the sub-message \( (x_i) = 7406080 \):
- Compute \( r \) and \( s \) (in the sender’s side);
  - \( r=a^\text{Ke} \mod P \)
  - \( r=8^7 \mod 1753 \)
  - \( r=2097152 \mod 1753=564 \)

\[
\begin{align*}
  s &= ((x-d*r)*\text{Ke}^{-1}) \mod (p-1) \\
  \text{Ke}^{-1} &= \text{the inverse of the } \text{Ke, } \text{Ke}^{-1}=751 \\
  s &= ((7406080-12*564)*751) \mod (p-1) \\
  s &= (7406080-6768)*751 \mod 1752 \\
  s &= 556883312 \mod 1752=88
\end{align*}
\]

With each sub-message \( (x_i) \) there is a new \( s \) value, but \( r \) is the same.

```
x_1=7406080  r=564  s=88
```

The third party will check the validity by the verification process as follow:-

The third party will compute \( T_g \) and \( T_y; \)
\[
\begin{align*}
  T_g &= (B^r \mod p) \\
  T_g &= (1026^{564} \mod 1753) \equiv 310 \\
  T_y &= a^7 \mod p \\
  T_y &= (7406080^{7}) \mod 1753 = 310
\end{align*}
\]
Since $T_2 = T_3$, then the digital signature of the $x_1$ is valid.

And so on for the $x_2$, $x_3$, $x_4$, $x_5$, and $x_6$, the computations in the above will be repeated six times for each DigCur with the same $p$, $a$, $d$, $B$, $ke$ and $ke^{-1}$.

In our system the template of the DigCur will be as follow:

| Serial number | Date of Issue | Currency class | Issuer | H_code (x): $(x_1, x_2, x_3, x_4, x_5, x_6)$ | Digital signature: $(p, a, B)$ | $(r, s_1, s_2, s_3, s_4, s_5, s_6)$ | Owner ID |
|---------------|---------------|----------------|--------|---------------------------------|---------------------------------|---------------------------------|----------|
|               |               |                |        |                                |                                 |                                 |          |

We used the mechanism of partitioning because of the problem of very large integer that our system can’t solve this problem.

For only one DigCur of class (50DigCur) will be as follow:

| Serial number | Date of Issue | Currency class | Issuer | H_code (x): $(x_1, x_2, x_3, x_4, x_5, x_6)$ | Digital signature: $(p, a, B)$ | $(r, s_1, s_2, s_3, s_4, s_5, s_6)$ | Owner ID |
|---------------|---------------|----------------|--------|---------------------------------|---------------------------------|---------------------------------|----------|
| 432432        | 14/4/2018     | 50DigCur       | ICB    | 710200e37d9909fdbe467bf41afec397 | 233432                         | 121567                          | Person1 |

For each $s_i$ the third party must compute $T_g$ and $T_y$, and so on for all the digital banknotes (DigCurs) that will be transmitted from Person1 to Person 2. In reality the main role of the third party (the central bank) is to check the hash code (MD5) because it only knows the secret key by merging the serial number with the secret key and then compute the hash code. The process of digital signature it is possible to be in the receiver side, but it is good to be accomplished in the third party’s servers (high speed computer). For each digital banknote the Owner ID will be updated to be as a receiver ID immediately after checking the hash code and the digital signature, in this case it will be Person 2, also there is a need for updating the copy of this digital banknote in the central database to make the IDs matching in the next transaction is possible.

For testing; suppose there are four DigCurs sent by per1 to per2 as shown below:

**First DigCur**

| Serial number | Date of Issue | Currency class | Issuer | H_code (x): $(x_1, x_2, x_3, x_4, x_5, x_6)$ | Digital signature: $(p, a, B)$ | $(r, s_1, s_2, s_3, s_4, s_5, s_6)$ | Owner ID |
|---------------|---------------|----------------|--------|---------------------------------|---------------------------------|---------------------------------|----------|
| 233432        | 21/11/2018    | 1DigCur        | ICB    | c10cc4df0f9f957c2964c0a102713ad88 | 1753 $a=8$ $B=1497$ $r=564$ | $s_1=152$ $s_2=67$ $s_3=1526$ $s_4=1668$ $s_5=759$ | Per1     |

**Second DigCur**

| Serial number | Date of Issue | Currency class | Issuer | H_code (x): $(x_1, x_2, x_3, x_4, x_5, x_6)$ | Digital signature: $(p, a, B)$ | $(r, s_1, s_2, s_3, s_4, s_5, s_6)$ | Owner ID |
|---------------|---------------|----------------|--------|---------------------------------|---------------------------------|---------------------------------|----------|
| 877613        | 21/11/2018    | 1DigCur        | ICB    | a9588d0e6814577937fe4b73b0e460c8 | 1753 $a=8$ $B=1497$ $r=564$ | $s_1=983$ $s_2=552$ $s_3=845$ $s_4=1641$ $s_5=1052$ | Per1     |

**Third DigCur**

| Serial number | Date of Issue | Currency class | Issuer | H_code (x): $(x_1, x_2, x_3, x_4, x_5, x_6)$ | Digital signature: $(p, a, B)$ | $(r, s_1, s_2, s_3, s_4, s_5, s_6)$ | Owner ID |
|---------------|---------------|----------------|--------|---------------------------------|---------------------------------|---------------------------------|----------|
| 655454        | 21/11/2018    | 1DigCur        | ICB    | fc09e4486aee5cf64133d7f13708d4f9 | 1753 $a=8$ $B=1497$ $r=564$ | $s_1=936$ $s_2=1694$ $s_3=1315$ $s_4=291$ $s_5=632$ | Per1     |
Digital Signature of the person (Person1) is:
\[ P=1753 \quad a=8 \quad B=1497 \quad r=564 \]
\[ s_1=540 \quad s_2=643 \quad s_3=1019 \quad s_4=220 \quad s_5=637 \quad s_6=846 \]

Per1

The secret key of the central bank (Sec_key) in all cases must be “test1978”, the sender’s private key is (d=27), Ke=7, and the sender’s public key is (p=1753, a=8, B=1497).

The first DigCur will be accepted because the H_code is valid and the digital signature for each \( s_i \) is valid also (\( T_g \) equal to \( T_y \)).

The second DigCur will be rejected because the H_code is not valid although the digital signature is valid.

The third DigCur will be rejected because the digital signature is not valid (unauthorized person) although the H_code is valid.

The fourth DigCur will be rejected because the system when checked it in the central database did not find this DigCur subordinate to Per1 (ID matching).

7- Conclusions

This paper aims to propose a method for generating a digital currency that is useful for daily financial transactions and make it as an alternative of the traditional currencies. Also this paper may help the central banks to issue their own digital currency. The proposed digital currency (DigCur) is digital currency not virtual currency because it is generated by the computer program with a full control by the central bank, whereas the central bank is the third party that will be responsible for verifying the process of the digital money transaction. The digital moneys are units called as digital banknotes. This work generates a digital currency that is difficult to counterfeit as in the case of paper banknotes. The problem of double spending is solved by the central bank by checking the user identifier and makes the required update directly after finishing the transaction process. DigCur gives a control on the money transactions by using a centralized database. Any centralized digital currency such as DigCur will reduce the concern from decentralized currencies, and this will protect the authority of the central banks. The hash code that is generated by a hash function is very helpful because of its uniqueness, whereas for each digital banknote (DigCur) there is a specific hash code. Also in this work we apply the digital signature to prevent the sender from denying the process of DigCur transfer. Also with some tests is possible to conclude the proposed method is good and novel idea for generating a centralized digital currency.

References

[1] Aumasson J., “Serious Cryptography: A Practical Introduction to Modern encryption”, No Starch Press, Inc., 2018.
[2] Berentsen A., Schar F., “A Short Introduction to the World of Cryptocurrencies”, Federal Reserve Bank of St. Louis Review, First Quarter 2018, https://doi.org/10.20955/r.2018.1-16.
[3] Chuen D., “Handbook of Digital Currency”, 1st Edition, Elsevier Inc, 2015.
[4] Elgamal T., “A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms”, IEEE Transactions on Information Theory, 31 (4), 469-472, 1985.
[5] Ferguson N., Schneier B., Kohno T., “Cryptography Engineering: Design Principles and Practical Applications”, Wiley Publishing, Inc., 2010.
[6] Law L., Sabett S., Solinas J., “How to Make a Mint: The Cryptography of Anonymous Electronic Cash”, American University Law Review, Vol. 46 no.4, 1997.
[7] Menezes A. J., Van Oorschot P. C., Vanstone S. A., “Handbook of Applied Cryptography”, CRC Press, 1996.
[8] Nakamoto S., “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2008, www.bitcoin.org/bitcoin.pdf
[9] Paar C., Pelzl J., “Understanding Cryptography”, Springer-Verlag Berlin Heidelberg, 2010.
[10] Redzovic M., Novakovic J., “The Impact of Virtual Money on E-commerce”, International Scientific Conference on ICT and E-business Related Research, 2016.
[11] Rivest R.L., “The MD5 message-digest algorithm”, Internet Request for Comments 1321, 1992, https://www.ietf.org/rfc/rfc1321.txt
[12] Stallings W., “Cryptography and Network Security: Principles and Practice”, Sixth Edition, Pearson Education, Inc., 2014.
[13] Tar A., “Digital Currencies vs. Cryptocurrencies, Explained”, 2017. https://cointelegraph.com/explained/digital-currencies-vs-cryptocurrencies-explained
[14] Thomas C.G., Jose R. T., “A Comparative Study on Different Hashing Algorithms”, International Journal of Innovative Research in Computer and Communication Engineering, 3(7), 2015.
[15] Valdes-Benavides R. A., Hernandez-Verme P. L., “Virtual Currencies, Micropayments and Monetary Policy: Where Are We Coming from and Where Does the Industry Stand?”, Journal of Virtual Worlds Research, 7(3) Lantern (2), August 2014.
[16] “What is Bitcoin Double Spending”, https://www.bitcoin.com/info/what-is-bitcoin-double-spending, 2017.