source

Digital Cash

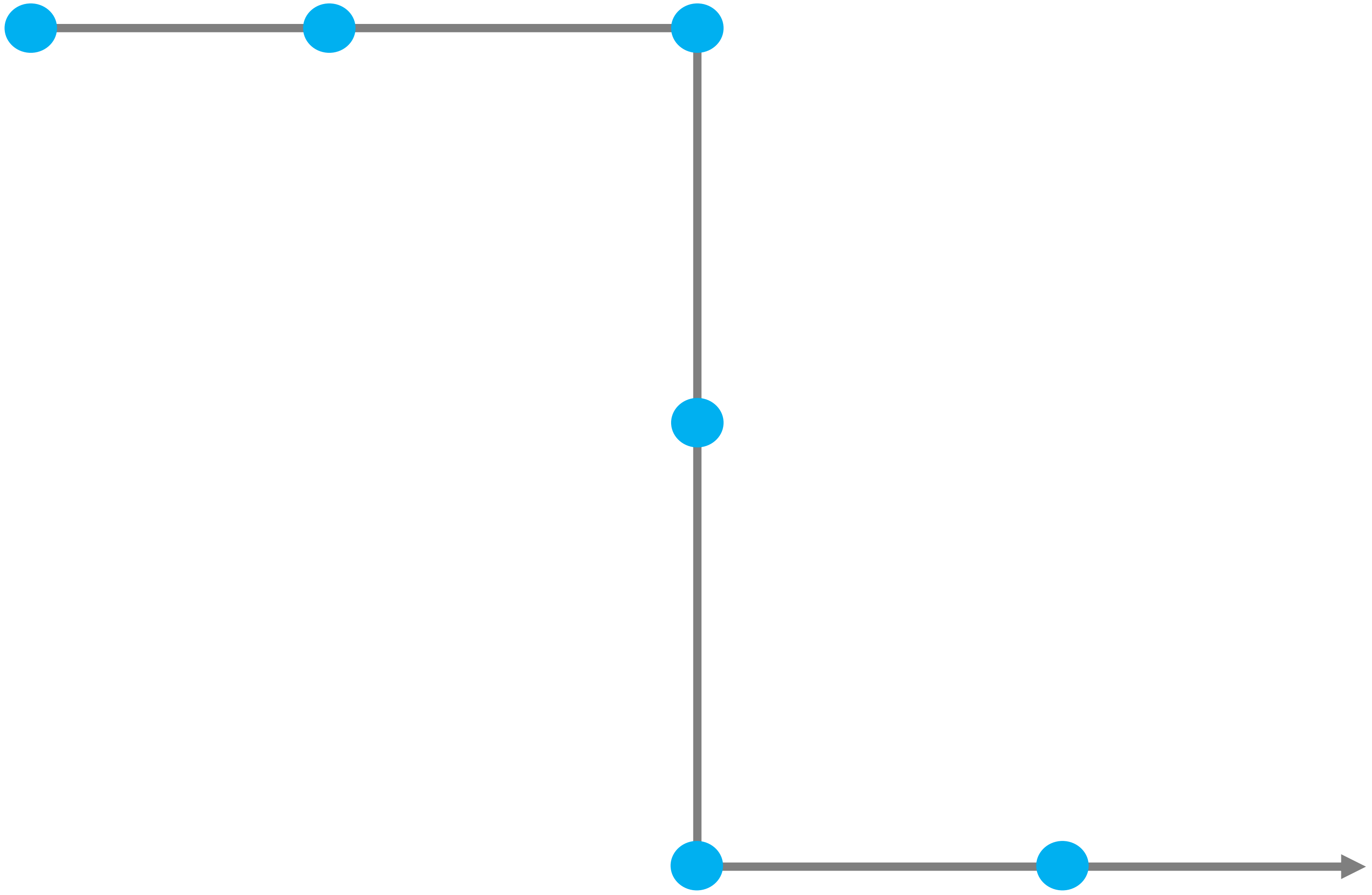
Cryptography – fall semester 2024,2025

Professor: Ziba Eslami

Sajjad Ranjbar Yazdi

Table of contents

Setting the stage for **Digital Economies**



Setting the stage for **Digital Economies**

Barter & Exchange

Setting the stage for **Digital Economies**

Barter & Exchange

"If you give me Fish, then I will provide you Wheat tomorrow"

Setting the stage for **Digital Economies**

Barter & Exchange

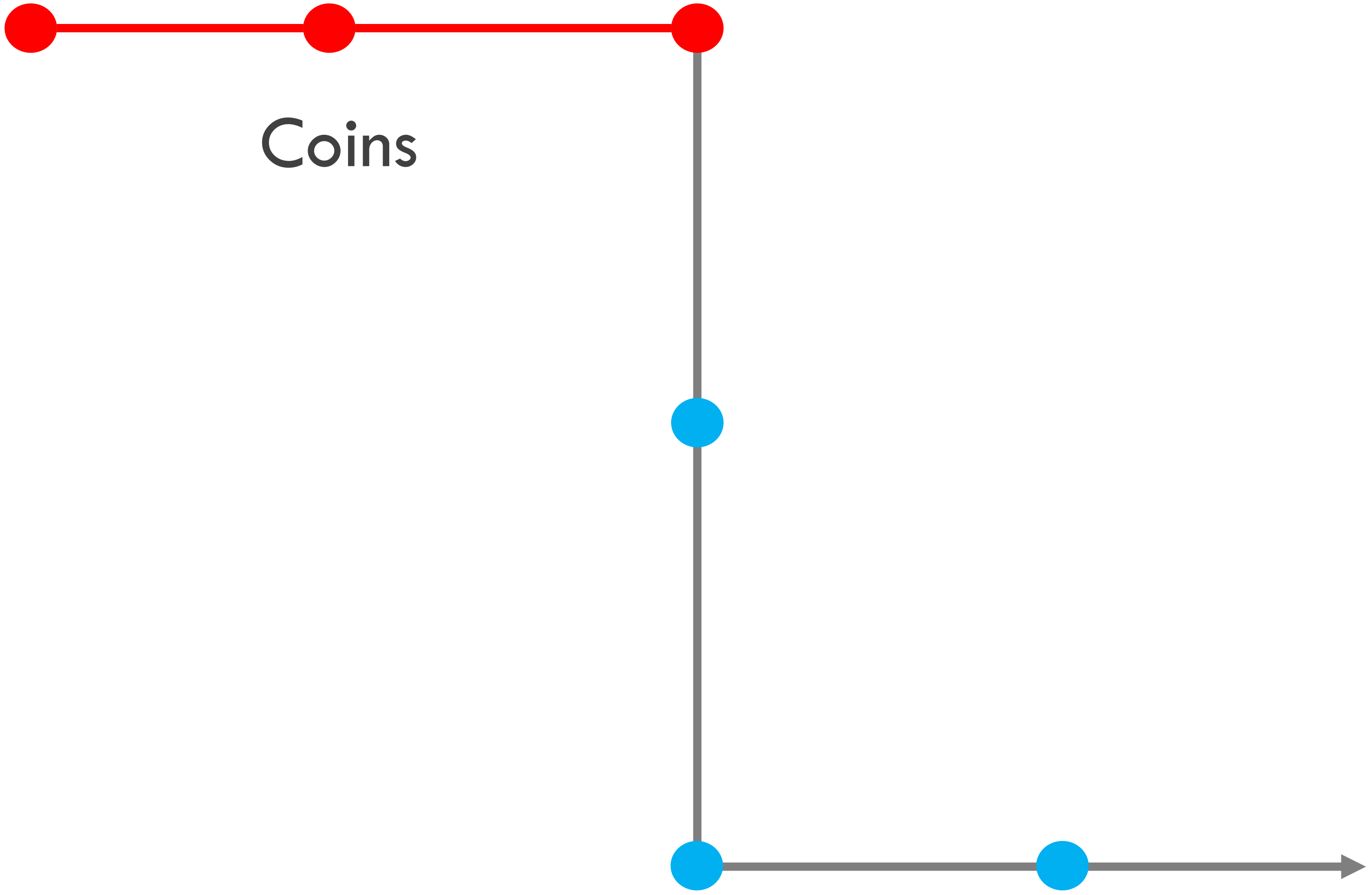
Credit

"If you give me Fish, then I will provide you Wheat tomorrow"

Setting the stage for **Digital Economies**

Precious elements

Setting the stage for **Digital Economies**



Setting the stage for **Digital Economies**

Coins

First

Lydian Lion

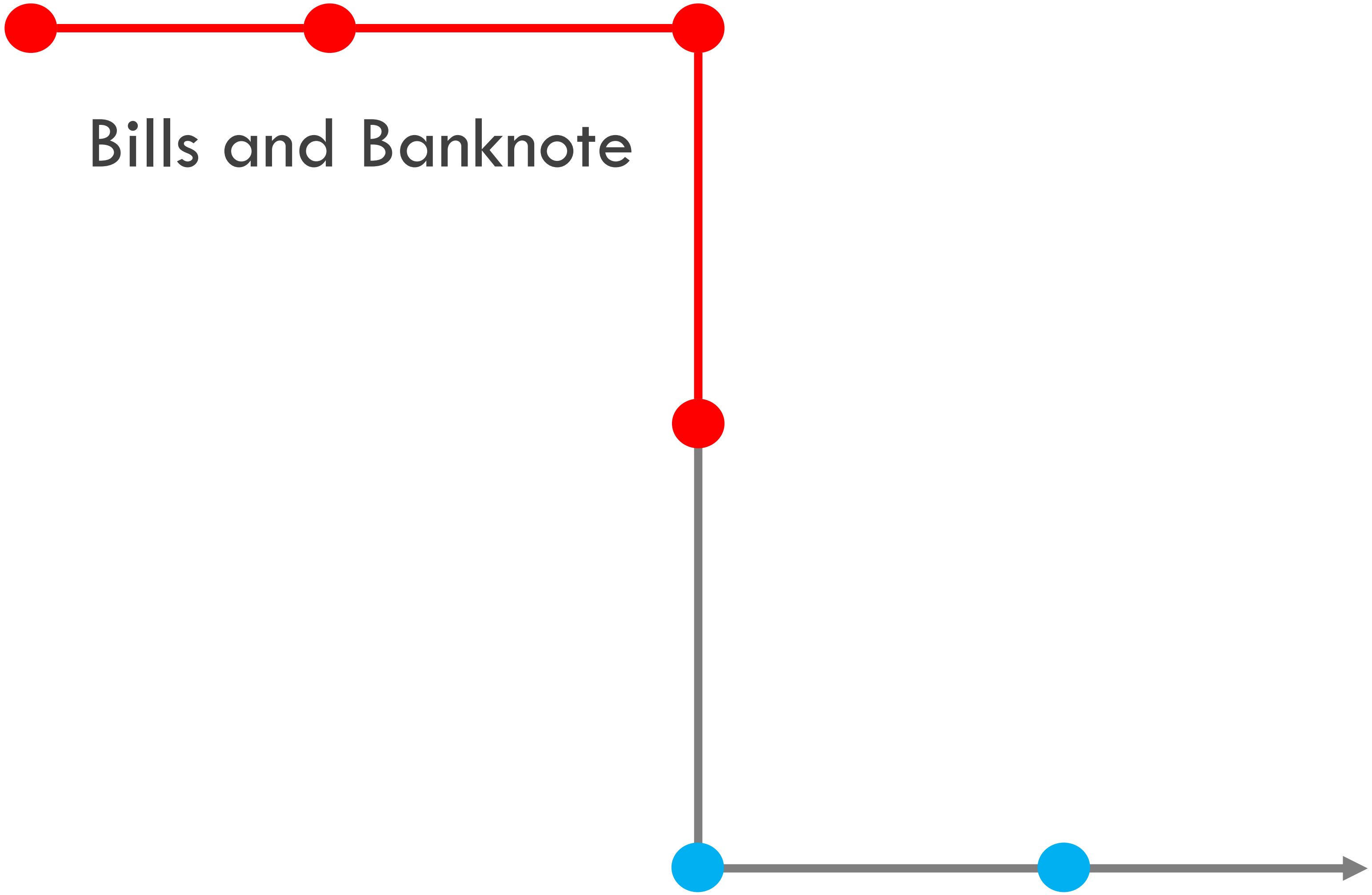
Setting the stage for **Digital Economies**

Coins

First

Lydian Lion

Setting the stage for **Digital Economies**



Setting the stage for **Digital Economies**

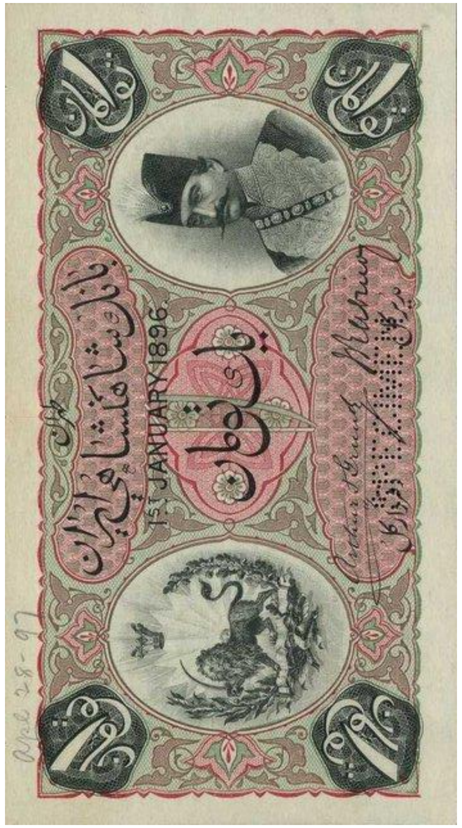
Bills and Banknote

Jiaozi

First

Setting the stage for **Digital Economies**

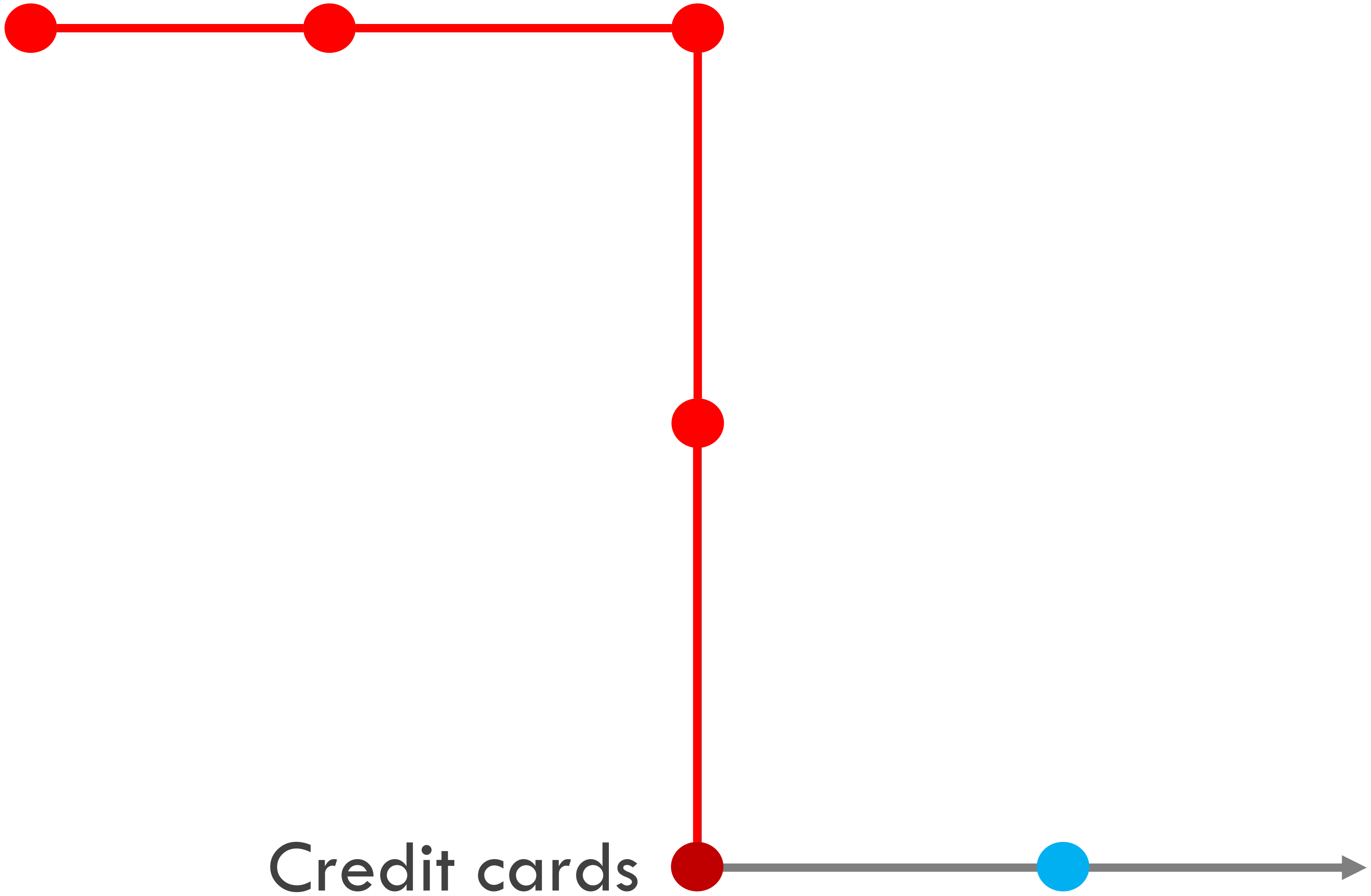
Bills and Banknote



Jiaozi

First

Setting the stage for **Digital Economies**



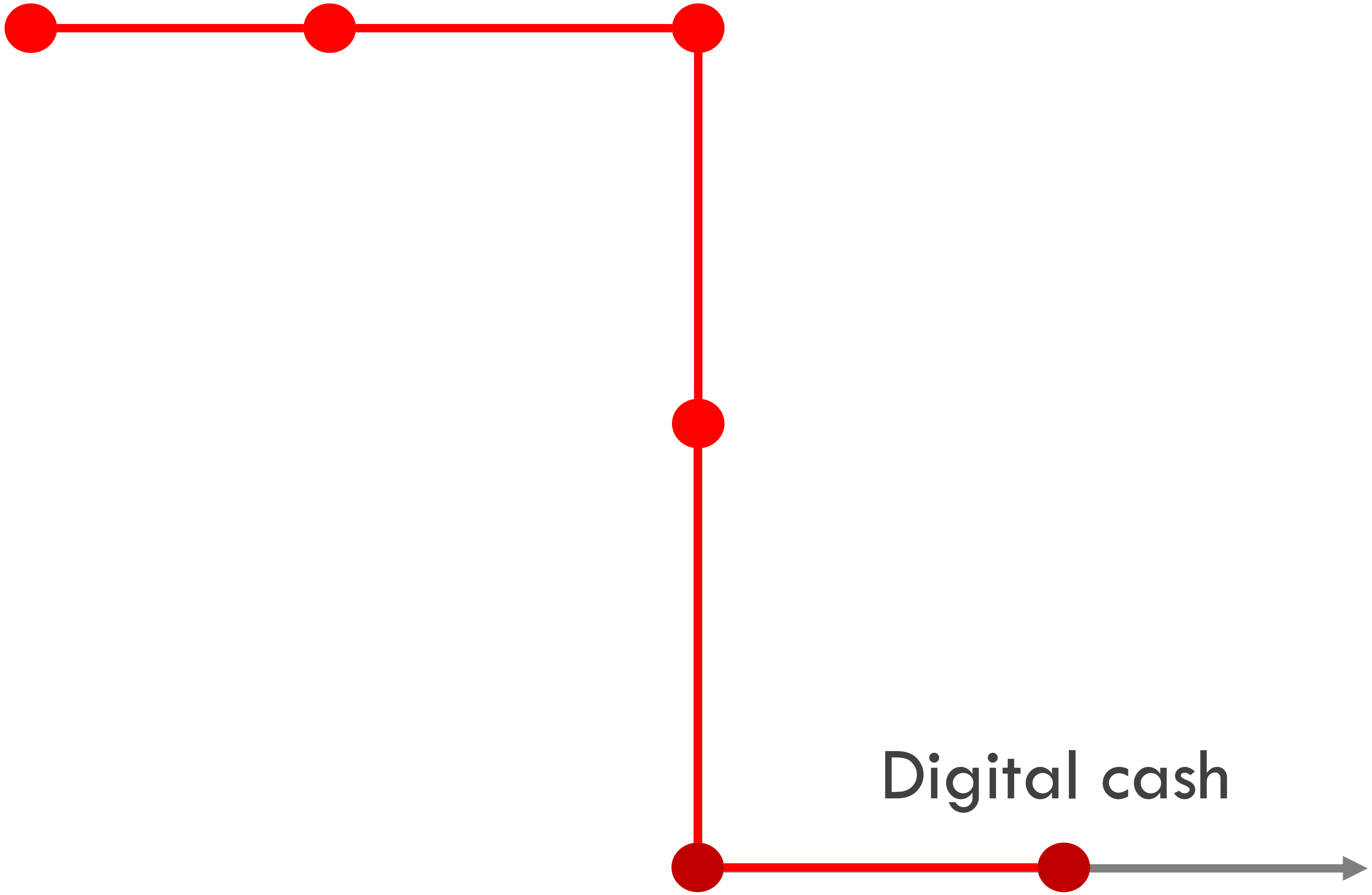
Setting the stage for **Digital Economies**

Diner's club card

First

Credit cards

Setting the stage for **Digital Economies**



Setting the stage for **Digital Economies**

It took several **millennia** from the first coin to the first banknote

Setting the stage for **Digital Economies**

It took several **millennia** from the first coin to the first banknote

It took several **centuries** from the first banknote to the first credit card

Setting the stage for Digital Economies

It took several **millennia** from the first coin to the first banknote

It took several **centuries** from the first banknote to the first credit card

It took only a few **decades** from the first credit card to digital cash

Setting the stage for **Digital Economies**

Science is advancing at a very fast pace. With this outlook, perhaps in a few years, everything related to our business will change. Therefore, it is incumbent upon us to move to the edge of science.

Digital cash system

Digital cash system

Participants

Digital cash system

Participants

Bank

Digital cash system

Participants

Spender

Bank

Digital cash system

Participants

Merchant

Spender

Bank

Bank

Issuance: The bank issues digital cash in Central Bank Digital Currencies (CBDCs) or as private digital money like stablecoins.

Verification: It verifies the authenticity of digital transactions to prevent fraud and counterfeiting.

Security: Banks use cryptographic measures to secure transactions and prevent unauthorized access or double-spending issues.

Regulation: Acts as a regulator, ensuring compliance with anti-money laundering (AML) and know-your-customer (KYC) norms.

Spender

User of Digital Cash: The spender uses digital cash to pay for goods or services. **Authentication:** They must authenticate themselves, often through digital wallets or other secure methods. Transaction **Initiation:** Initiates transactions by transferring digital cash to merchants, ensuring they have the required balance. **Privacy:** In certain systems, spenders can remain anonymous, depending on the cryptographic and policy frameworks in use.

Merchant

Acceptance of Payments: Merchants receive payments in digital cash for goods or services provided.

Integration with Payment Systems: They must integrate with digital payment infrastructure, like wallets and point-of-sale (POS) systems.

Settlement Requests: Merchants request settlement from the bank or payment intermediary, converting digital cash into their desired form (e.g., fiat currency).

Compliance: Ensure compliance with taxation and regulatory frameworks for digital transactions.

Initialization a Digital cash system

Initialization a Digital cash system

Choose large prime number **p**, such that **q**=(p-1)/2 is also prime.

Initialization a Digital cash system

Choose large prime number **p**, such that **q**=(p-1)/2 is also prime.

Let **g** be the square of a primitive root mod p.

Initialization a Digital cash system

Choose large prime number **p**, such that **q**=(p-1)/2 is also prime.

Let **g** be the square of a primitive root mod p.

```
def find_square_of_primitive_root(p):
    g0 = primitive_root(p) # imported from sympy
    # Calculate the square of the primitive root modulo p
    g = pow(g0, 2, p)
    return g
```

Initialization a Digital cash system

Choose large prime number **p**, such that **q**=(p-1)/2 is also prime.

Let **g** be the square of a primitive root mod p.

$$g^{k1} \equiv g^{k2} \pmod{p} \iff k_1 \equiv k_2 \pmod{q}$$

Initialization a Digital cash system

Choose large prime number **p**, such that **q**=(p-1)/2 is also prime.

Let **g** be the square of a primitive root mod p.

Two secret random exponent!

$$g^{k1} \equiv g^{k2} \pmod{p} \iff k_1 \equiv k_2 \pmod{q}$$

Initialization a Digital cash system

Choose large prime number **p**, such that **q**=(p-1)/2 is also prime.

Let **g** be the square of a primitive root mod p.

Two secret random exponent!

$$g^{k1} \equiv g^{k2} \pmod{p} \iff k_1 \equiv k_2 \pmod{q}$$

g_1, g_2

Initialization a Digital cash system

We need two public hash functions.

Initialization a Digital cash system

We need two public hash functions.

H: takes a 5-tuple integers and outputs an integer mod q.

Initialization a Digital cash system

We need two public hash functions.

H: takes a 5-tuple integers and outputs an integer mod q.

```
def hash_H(input_tuple):
    if len(input_tuple) != 5:
        raise ValueError("Input must be a 5-tuple of integers.")
    input_bytes = ','.join(map(str, input_tuple)).encode('utf-8’)
    digest = Hash(SHA256(), backend=default_backend())
    digest.update(input_bytes)
    hash_digest = digest.finalize()
    return int.from_bytes(hash_digest, 'big') % q
```

Initialization a Digital cash system

We need two public hash functions.

H: takes a 5-tuple integers and outputs an integer mod q.

H_0: takes a 4-tuple integers and outputs an integer mod q.

Initialization a Digital cash system

We need two public hash functions.

H: takes a 5-tuple integers and outputs an integer mod q.

H_0: takes a 4-tuple integers and outputs an integer mod q.

```
def hash_H0(input_tuple):
    if len(input_tuple) != 4:
        raise ValueError("Input must be a 4-tuple of integers.")
    input_bytes = ','.join(map(str, input_tuple)).encode('utf-8’)
    digest = Hash(SHA256(), backend=default_backend()) digest.update(input_bytes)
    hash_digest = digest.finalize()
    return int.from_bytes(hash_digest, 'big') % q
```

Initialization a Digital cash system

So after initialization steps we have:

Initialization a Digital cash system

So after initialization steps we have:

$$p,\ q,\ g,\ g_1,\ g_2,\ H,\ H_0$$

The Bank

The Bank

The bank choose its secret identity number “**x**”

The Bank

The bank choose its secret identity number “**x**”

$h \equiv g^x \pmod{p}$

The Bank

The bank choose its secret identity number “**x**”

$h \equiv g^x \pmod p$

The number “**h**” is made public and identifies the bank

The Spender

The **Spender**

The spender choose its secret identity number “**u**”

The **Spender**

The spender choose its secret identity number "**u**"

$\mathbf{I} \equiv g_1^{u} \pmod{p}$

The **Spender**

The spender choose its secret identity number “**u**”

$\mathbf{I} \equiv g_1^{u} \pmod{p}$

Account number of the spender

The Spender

The spender choose its secret identity number “**u**”

$\mathbf{I} \equiv g_1^{u}\ (\mathrm{mod}\ p)$

Account number of the spender

The spender sends “**I** “ to the bank and the bank stores it with other information like name, address, etc.

The Merchant

The Merchant

The Merchant choose an identification number “**M**”

The Merchant

The Merchant choose an identification number "**M**"

The Merchant registers "M" with the bank

Creating a Coin

Creating a Coin

The Spender contacts the bank, asking for a coin

Creating a Coin

The Spender contacts the bank, asking for a coin

The bank requires proof of identity, just as when someone is withdrawing classical cash from an account!

Creating a Coin

The Spender contacts the bank, asking for a coin

The bank requires proof of identity, just as when someone is withdrawing classical cash from an account!

All coins in the present scheme have the same value!

Creating a Coin

We can present a coin with 6-tuple of numbers

$$(A, B, z, a, b, r)$$

Creating a Coin

(A, B, z, a, b, r)

Creating a Coin

(A, B, z, a, b, r)

The Bank chooses a random number “ω”

Creating a Coin

(A, B, z, a, b, r)

The Bank chooses a random number "ω"

(ω is a different number for each coin)

Creating a Coin

(A, B, z, a, b, r)

The Bank chooses a random number "ω"

(ω is a different number for each coin)

$$g_{\omega} \equiv g^{\omega} \pmod{p}$$

Creating a Coin

(A, B, z, a, b, r)

The Bank chooses a random number "ω"

(ω is a different number for each coin)

$$g_\omega \equiv g^{\omega} \pmod{p}$$

$$\beta \equiv (I\, g_2)^{\omega} \pmod{p}$$

Creating a Coin

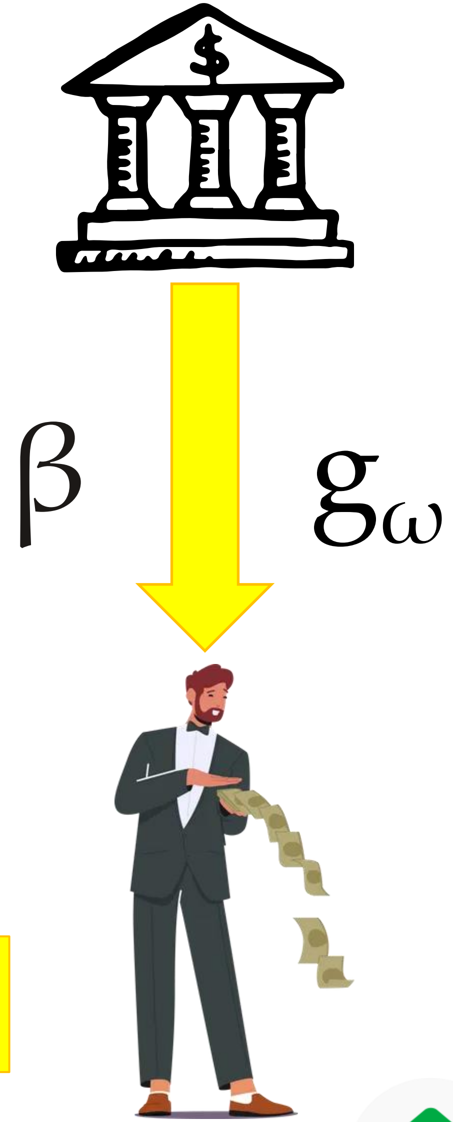
(A, B, z, a, b, r)

The Bank chooses a random number "ω"

(ω is a different number for each coin)

$$g_\omega \equiv g^{\omega} \pmod{p}$$

$$\beta \equiv (I\, g_2)^{\omega} \pmod{p}$$



The bank sends β and g_ω to the spender

Creating a Coin

(A, B, z, a, b, r)

The Spender chooses 5 random integers.

Creating a Coin

(A, B, z, a, b, r)

The Spender chooses 5 random integers.

$(s, x_1, x_2, \alpha_1, \alpha_2)$

Creating a Coin

(A, B, z, a, b, r)

The Spender chooses 5 random integers.

$(s, x_1, x_2, \alpha_1, \alpha_2)$

$A \equiv (Ig2)^s \pmod p$

Creating a Coin

(A, B, z, a, b, r)

The Spender chooses 5 random integers.

$(s, x_1, x_2, \alpha_1, \alpha_2)$

Coins with A = 1 are not allowed

$A \equiv (Ig2)^s \pmod{p}$

Creating a Coin

(A, B, z, a, b, r)

The Spender chooses 5 random integers.

$(s, x_1, x_2, \alpha_1, \alpha_2)$

This can happen in only two ways. One is when $s \equiv 0 \pmod{q}$, so we require $s \not\equiv 0$

Coins with $A = 1$ are not allowed

$A \equiv (Ig2)^s \pmod{p}$

Creating a Coin

(A, B, z, a, b, r)

The Spender chooses 5 random integers.

$(s, x_1, x_2, \alpha_1, \alpha_2)$

$$A \equiv (Ig2)^s \pmod{p}$$

$$B \equiv g_1^{x1} g_2^{x2} \pmod{p}$$

Creating a Coin

(**A**, **B**, **z**, **a**, **b**, **r**)

The Spender chooses 5 random integers.

$(s, x_1, x_2, \alpha_1, \alpha_2)$

$A \equiv (Ig2)^s \pmod p$

$a \equiv g_{\omega}^{\alpha 1} g^{\alpha 2} \pmod p$

$B \equiv g_1^{x1} g_2^{x2} \pmod p$

Creating a Coin

(A, B, z, a, b, r)

The Spender chooses 5 random integers.

$(s, x_1, x_2, \alpha_1, \alpha_2)$

$A \equiv (Ig2)^s \pmod p$

$a \equiv g_{\omega}^{\alpha 1} g^{\alpha 2} \pmod p$

$B \equiv g_1^{x1} g_2^{x2} \pmod p$

$b \equiv \beta^{s\alpha 1} A^{\alpha 2} \pmod p$

Creating a **Coin**

(A, B, z, a, b, r)

The Spender chooses 5 random integers.

$(s, x_1, x_2, \alpha_1, \alpha_2)$

$A \equiv (Ig2)^s \pmod{p}$

$a \equiv g_{\omega}^{\alpha 1} g^{\alpha 2} \pmod{p}$

$z \equiv z'^s \pmod{p}$

$B \equiv g_1^{x1} g_2^{x2} \pmod{p}$

$b \equiv \beta^{s\alpha 1} A^{\alpha 2} \pmod{p}$

Creating a Coin

(A, B, z, a, b, r)

Creating a Coin

(A, B, z, a, b, r)

Creating a Coin

(A, B, z, a, b, r)

Creating a Coin

(**A**, **B**, **z**, **a**, **b**, **r**)

$$c \equiv \alpha_1^{-1} H(A, B, z, a, b) \pmod{p}$$

Creating a Coin

(A, B, z, a, b, r)

$$c \equiv \alpha_1^{-1} H(A, B, z, a, b) \pmod{p}$$

$$c_1 \equiv cx + \omega \pmod{p}$$

Creating a **Coin**

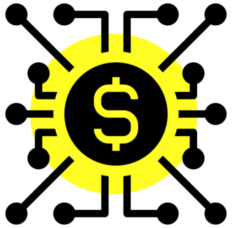
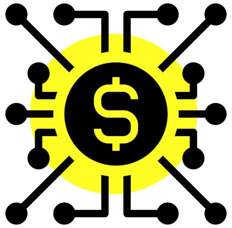
(A, B, z, a, b, r)

$$c \equiv \alpha_1^{-1} H(A, B, z, a, b) \pmod{p}$$

$$c_1 \equiv cx + \omega \pmod{p}$$

$$r \equiv \alpha_1 c_1 + \alpha_2 \pmod{p}$$

Creating a Coin

 (A, B, z, a, b, r) 

Spending a Coin

Merchant

Spender

Spending a Coin

Merchant

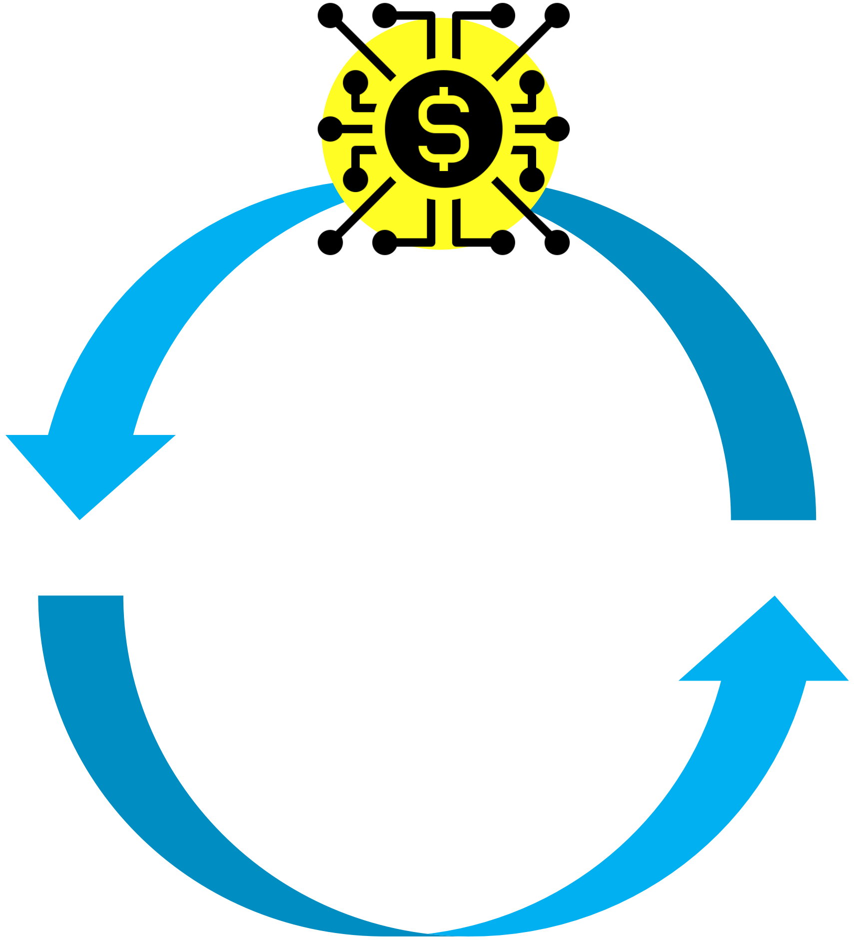
Goods or services

Spender

Spending a Coin

(A, B, z, a, b, r)

Merchant



Goods or services

Spender

Spending a Coin

(A, B, z, a, b, r)

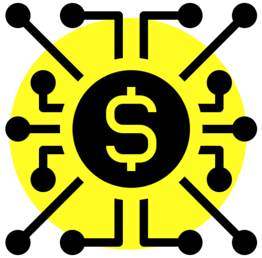
Merchant

$$g^r \equiv a\, h^{H(A,\, B,\, z,\, a,\, b)} \pmod p$$

$$A^r \equiv b\, z^{H(A,\, B,\, z,\, a,\, b)} \pmod p$$

Spending a **Coin**

$$(A, B, z, a, b, r)$$



Merchant

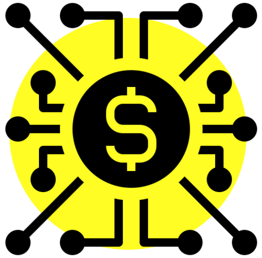
$$g^r \equiv a\, h^{H(A,\, B,\, z,\, a,\, b)} \pmod{p}$$

$$A^r \equiv b\, z^{H(A,\, B,\, z,\, a,\, b)} \pmod{p}$$

The merchant makes sure of the authenticity of the coin

Spending a Coin

(A, B, z, a, b, r)

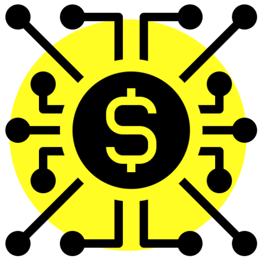


Merchant

$$d \equiv H_0(A, B, M, t)$$

Spending a Coin

(A, B, z, a, b, r)



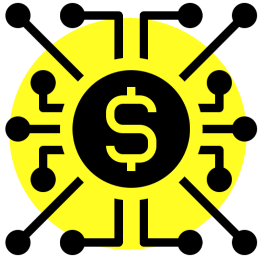
Merchant

$$d \equiv H_0(A, B, M, t)$$

t is a number representing date and time of transaction.

Spending a Coin

(A, B, z, a, b, r)



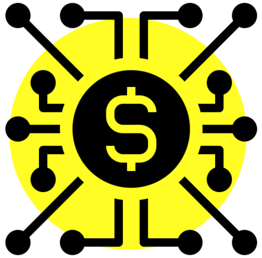
d

Merchant

Spender

Spending a Coin

$$(A, B, z, a, b, r)$$



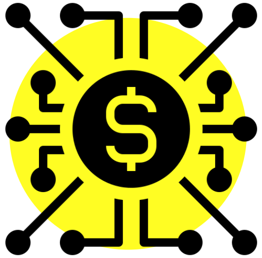
$$r_1 \equiv dus + x_1 \pmod q$$

$$r_2 \equiv ds + x_2 \pmod q$$

Spender

Spending a Coin

(A, B, z, a, b, r)



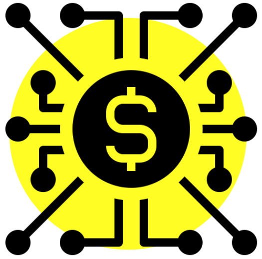
$r_1\ r_2$

Merchant

Spender

Spending a Coin

(A, B, z, a, b, r)



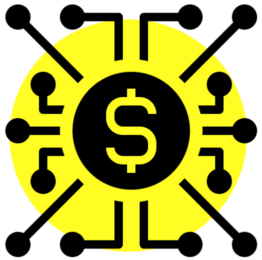
Merchant

$$\begin{cases} \text{if } g_1^{r1}\, g_2^{r2} \equiv A^d B \pmod{p} \\ \qquad \textbf{ACCEPT} \\ \text{else} \\ \qquad \textbf{REJECT} \end{cases}$$

Deposit a Coin in the bank

A few days after receiving the coin, the Merchant wants to deposit it in the Bank

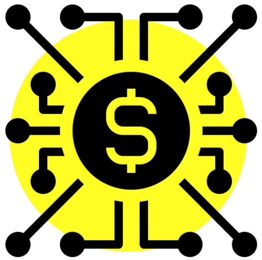
Deposit a Coin in the bank



A few days after receiving the coin, the Merchant wants to deposit it in the Bank

The Merchant submits the coin plus the triple of transaction

Deposit a Coin in the bank

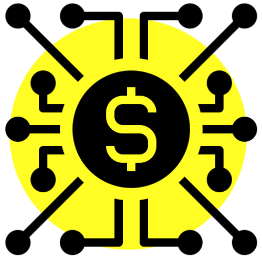


A few days after receiving the coin, the Merchant wants to deposit it in the Bank.

The Merchant submits the coin plus the triple of transaction

(A, B, z, a, b, r)

Deposit a Coin in the bank



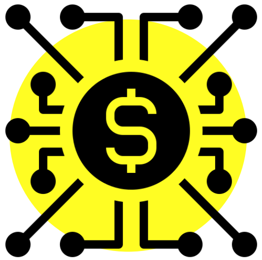
A few days after receiving the coin, the Merchant wants to deposit it in the Bank.

The Merchant submits the coin plus the triple of transaction

(A, B, z, a, b, r)

(r_1, r_2, d)

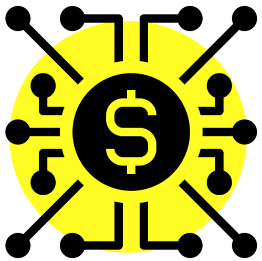
Deposit a Coin in the bank



Merchant

Bank

Deposit a Coin in the bank



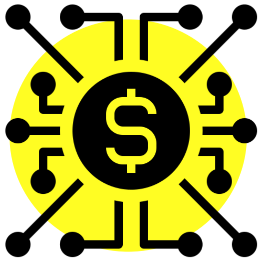
Merchant

(A, B, z, a, b, r)

(r_1, r_2, d)

Bank

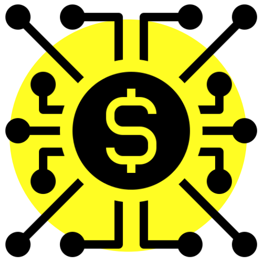
Deposit a Coin in the bank



The checks that the coin (A, B, z, a, b, r) has not been previously deposited.

Bank

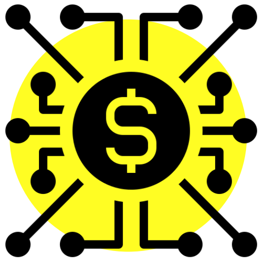
Deposit a Coin in the bank



Is coin valid?

Bank

Deposit a Coin in the bank



Is coin valid?

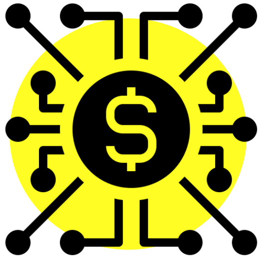
$$g^r \equiv a\, h^{H(A,\, B,\, z,\, a,\, b)} \pmod{p}$$

$$A^r \equiv b\, z^{H(A,\, B,\, z,\, a,\, b)} \pmod{p}$$

$$g_1^{\,r1}\, g_2^{\,r2} \equiv A^d B \pmod{p}$$

Bank

Deposit a Coin in the bank



Is coin valid?

$$g^{r} \equiv a\, h^{H(A,\, B,\, z,\, a,\, b)} \pmod{p}$$

$$A^{r} \equiv b\, z^{H(A,\, B,\, z,\, a,\, b)} \pmod{p}$$

$$g_1^{\,r1}\, g_2^{\,r2} \equiv A^{d}B \pmod{p}$$

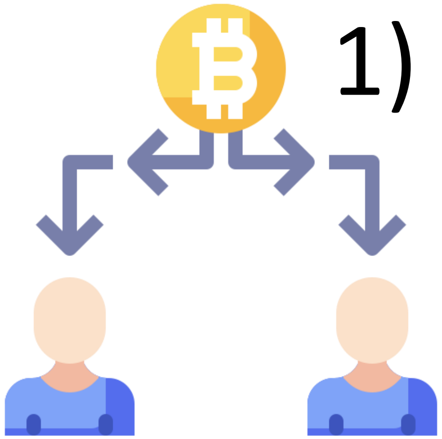
Bank

Deposit a Coin in the bank

coin is valid and the Merchant’s account is credited.

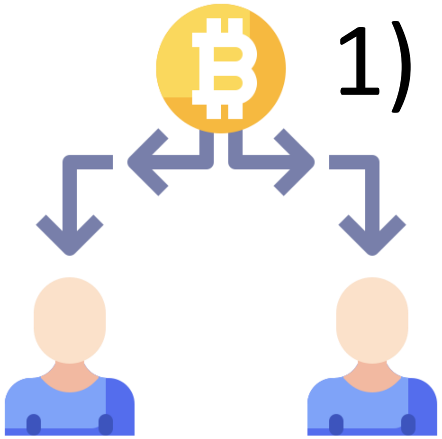
Fraud control

Fraud control



1) Double spending problem

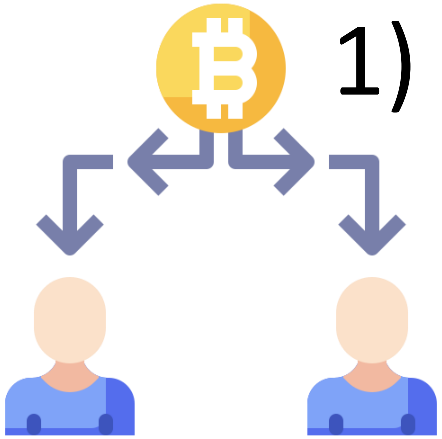
Fraud control



1) Double spending problem

The spender spends the coin twice, once with Merchant1 and once with Merchant2.

Fraud control



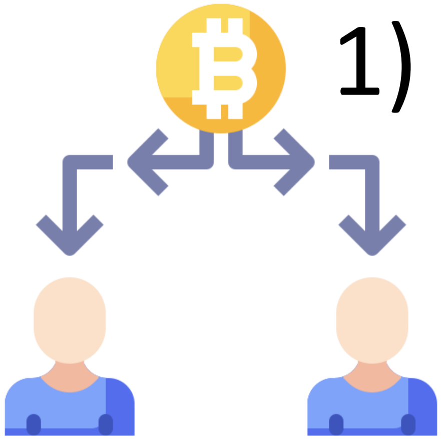
1) Double spending problem

The spender spends the coin twice, once with Merchant1 and once with Merchant2.

(A, B, z, a, b, r)
(r_1, r_2, d)

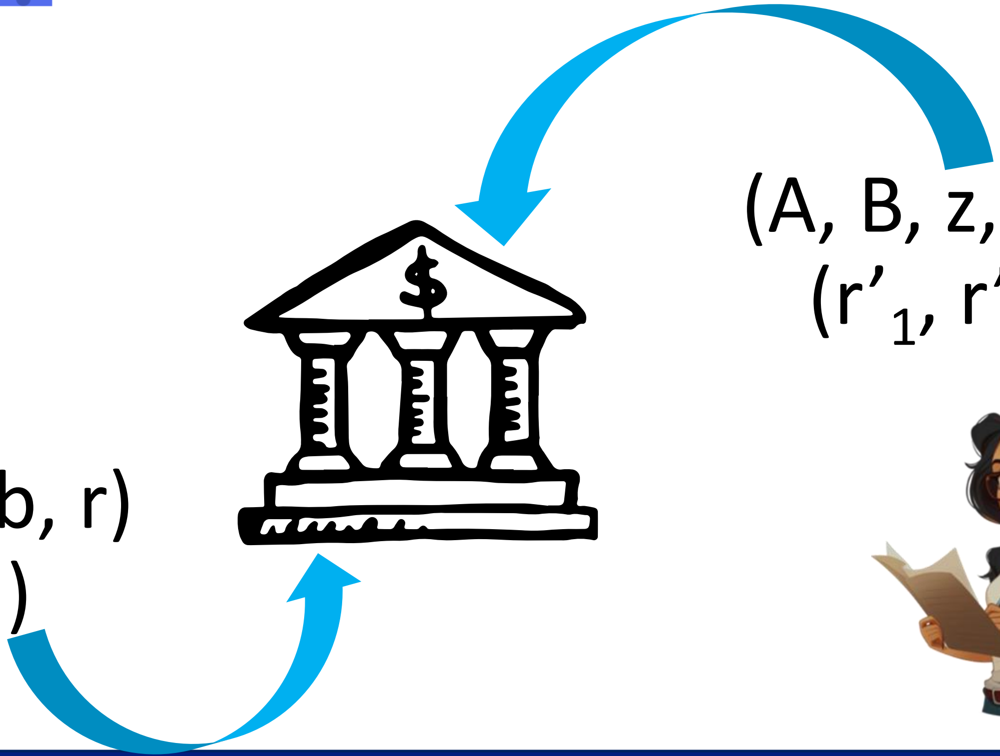
(A, B, z, a, b, r)
(r'_1, r'_2, d')

Fraud control



1) Double spending problem

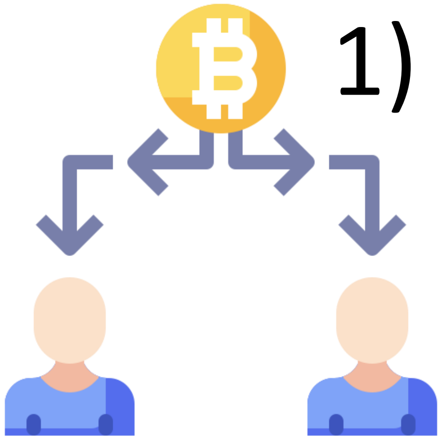
The spender spends the coin twice, once with Merchant1 and once with Merchant2.



(A, B, z, a, b, r)
(r'_1, r'_2, d')

(A, B, z, a, b, r)
(r_1, r_2, d)

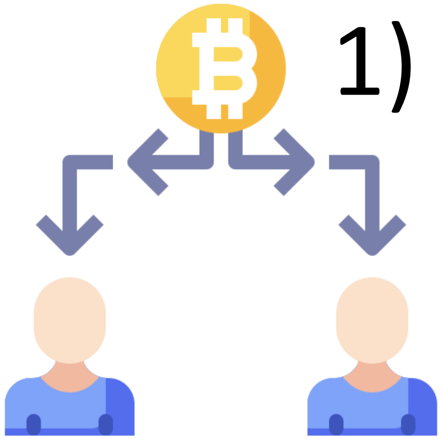
Fraud control



1) Double spending problem

(r_1, r_2, d) (r'_1, r'_2, d')

Fraud control



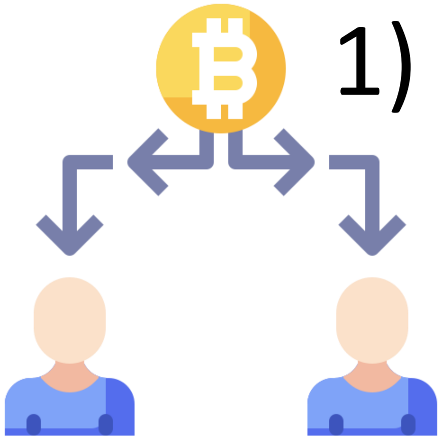
1) Double spending problem

(r_1, r_2, d) (r'_1, r'_2, d')

$$r_1 - r'_1 \equiv us(d-d') \pmod{p}$$

$$r_2 - r'_2 \equiv s(d-d') \pmod{p}$$

Fraud control



1) Double spending problem

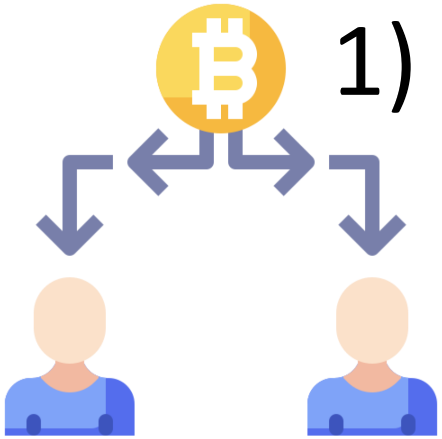
(r_1, r_2, d) (r'_1, r'_2, d')

$$r_1 - r'_1 \equiv us(d-d') \pmod{p}$$

$$r_2 - r'_2 \equiv s(d-d') \pmod{p}$$

$$u \equiv (r_1 - r'_1)(r_2 - r'_2)^{-1} \pmod{q}$$

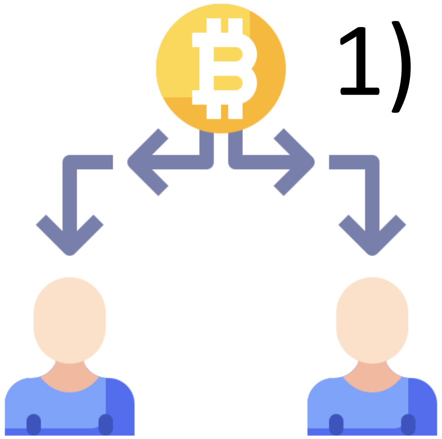
Fraud control



1) Double spending problem

$$\mathbf{I} \equiv g^{u}{}_{1}(\mathrm{mod}\ p)$$

Fraud control



1) Double spending problem

$$\mathbf{I} \equiv {g^{u}}_{1} (\mathrm{mod}\ p)$$

The spender who did this will be found

Fraud control

2) The Merchant tries to submitting the coin twice.

Fraud control

2) The Merchant tries to submitting the coin twice.

(r_1, r_2, d) (r'_1, r'_2, d')

Fraud control

2) The Merchant tries to submitting the coin twice.

(r_1, r_2, d) (r'_1, r'_2, d')

This is essentially **impossible** for the Merchant to do since it is complicated for the Merchant to produce numbers such that:

$$g_1^{r'1}\, g_2^{r'2} \equiv A^{d'}B \ (\mathrm{mod}\ p)$$

Fraud control

3) Someone tries to make an unauthorized coin

Fraud control

3) Someone tries to make an unauthorized coin

This requires finding numbers such that:

$g^r \equiv a\ h^{H(A, B, z, a, b)}$ and $A^r \equiv {}^{zH(A, B, z, a, b)}$

Fraud control

3) Someone tries to make an unauthorized coin

This requires finding numbers such that:
$g^r \equiv a\ h^{H(A,\ B,\ z,\ a,\ b)}$ and $A^r \equiv {}^{zH(A,\ B,\ z,\ a,\ b)}$

This is probably hard

Fraud control

3) Someone tries to make an unauthorized coin

This requires finding numbers such that:

$g^r \equiv a\ h^{H(A,\ B,\ z,\ a,\ b)}$ and $A^r \equiv {}^{zH(A,\ B,\ z,\ a,\ b)}$

This is probably hard

discrete logarithm problem

Fraud control

4) Someone working in the Bank tries to forge a coin

Fraud control

4) Someone working in the Bank tries to forge a coin

It is possible to make a coin that satisfies

$g^r \equiv a\ h^{H(A,\ B,\ z,\ a,\ b)}$

Fraud control

4) Someone working in the Bank tries to forge a coin

It is possible to make a coin that satisfies $g^r \equiv a\, h^{H(A,\, B,\, z,\, a,\, b)}$

However, since the Spender has kept u secret, the person in the bank cannot produce a suitable r_1

Fraud control

4) Someone working in the Bank tries to forge a coin

It is possible to make a coin that satisfies
$g^r \equiv a\ h^{H(A,\ B,\ z,\ a,\ b)}$
However, since the Spender has kept u secret, the person in the bank cannot produce a suitable r_1

Of course, this would be possible if s = 0 were allowed!

Flash back

(A, B, z, a, b, r)

$(s, x_1, x_2, \alpha_1, \alpha_2)$

This can happen in only two ways. One is when $s \equiv 0 \pmod q$, so we require $s \not\equiv 0$

$A \equiv (Ig2)^s \pmod p$

Coins with $A = 1$ are not allowed

Fraud control

4) Someone working in the Bank tries to forge a coin

It is possible to make a coin that satisfies

$g^r \equiv a\, h^{H(A,\, B,\, z,\, a,\, b)}$

However, since the Spender has kept u secret, the person in the bank cannot produce a suitable r_1

Of course, this would be possible if s = 0 were allowed!

This is one reason A = 1 is not allowed!!!

Fraud control

5) Someone steals the coin from the Spender and tries to spend it

Fraud control

5) Someone steals the coin from the Spender and tries to spend it

The first verification equation is still satisfied.

Fraud control

5) Someone steals the coin from the Spender and tries to spend it

The first verification equation is still satisfied.

but the thief does not know u and therefore will not be able to produce r_1, r_2 such that $g_1^{r1} g_2^{r2} \equiv A^{d'} B$

Anonymity

Anonymity in digital cash refers to the degree to which a user's identity and transactions are concealed when they interact with a digital currency system. Traditional cash transactions are inherently anonymous; they don't link to a user's identity unless explicitly tracked. In the digital realm, achieving a similar level of privacy is complex due to the technological infrastructure and regulatory frameworks involved!

Anonymity

Sender Anonymity: The sender's identity is hidden from the recipient or any third party.

Anonymity

Sender Anonymity: The sender's identity is hidden from the recipient or any third party.

Receiver Anonymity: The recipient's identity is concealed from the sender or others.

Anonymity

Sender Anonymity: The sender's identity is hidden from the recipient or any third party.

Receiver Anonymity: The recipient's identity is concealed from the sender or others.

Transaction Anonymity: The details of the transaction, including the parties involved and the amount, are obfuscated.

Anonymity

Technologies Enabling Anonymity

Anonymity

Technologies Enabling Anonymity

1. Zero-Knowledge Proofs (ZKPs)
2. Ring Signatures
3. Mixing Services
4. Blind Signatures

Technologies Enabling Anonymity

1. Zero-Knowledge Proofs (ZKPs)
2. Ring Signatures
3. Mixing Services
4. Blind Signatures

Bitcoin


DIGITAL * DECENTRALIZED * PEER TO PEER *
CRYPTOCURRENCY * BITCOIN

Bitcoin

Bitcoin is presented as a decentralized digital currency introduced by **Satoshi Nakamoto** in 2008 through a white paper titled *"Bitcoin: A Peer-to-Peer Electronic Cash System."*

Bitcoin

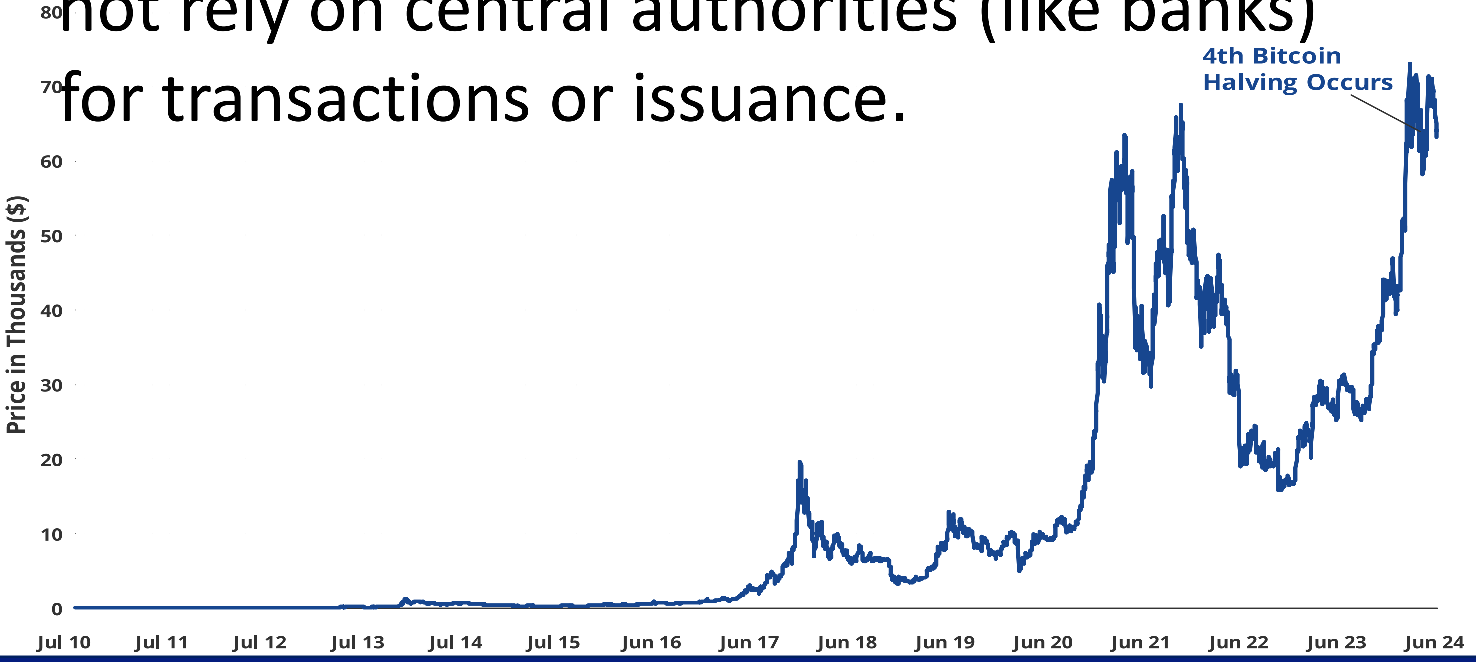
Bitcoin is presented as a decentralized digital currency introduced by **Satoshi Nakamoto** in 2008 through a white paper titled *"Bitcoin: A Peer-to-Peer Electronic Cash System."*

Unlike traditional currencies, Bitcoin does not rely on central authorities (like banks) for transactions or issuance.

Bitcoin

Bitcoin is presented as a decentralized digital currency introduced by **Satoshi Nakamoto** in 2008 through a white paper titled *"Bitcoin: A Peer-to-Peer Electronic Cash System."*

Unlike traditional currencies, Bitcoin does not rely on central authorities (like banks) for transactions or issuance.



Bitcoin

Key Cryptographic Principles in Bitcoin

Key Cryptographic Principles in Bitcoin

•**Public-Key Cryptography** cilbup esu snoticasnarT : gningis dna notiacfitinedi eruces rof syek etavirp dna

Bitcoin

Key Cryptographic Principles in Bitcoin

•**Public-Key Cryptography** cilbup esu snoticasnarT : gningis dna notiacfitinedi eruces rof syek etavirp dna
•**Hash Functions**: Cryptographic hash functions (like SHA-256 (ensure the integrity of data in transactions and are essential to the mining process

Bitcoin

Key Cryptographic Principles in Bitcoin

•**Public-Key Cryptography** cilbup esu snoticasnarT :
gningis dna notiacfitinedi eruces rof syek etavirp dna
•**Hash Functions**: Cryptographic hash functions (like
SHA-256 (ensure the integrity of data in transactions
and are essential to the mining process
•**Digital Signatures** fo ytictinehtua eht yfirev ot desU :
a fo renwo lutfhgir eht ylno taht gnirusne ,snoticasnart
ti dneps nac nioctiB

Bitcoin

Blockchain

Bitcoin

Blockchain

The blockchain is the backbone of Bitcoin and other cryptocurrencies. It is a distributed ledger that records all transactions in a transparent, tamper-proof, and decentralized manner. Transactions are grouped into blocks, which are linked sequentially in a chain using cryptographic hashes

Bitcoin

Block

Bitcoin

list of transactions

Block

Bitcoin

Block

list of transactions

hash of the previous block

Bitcoin

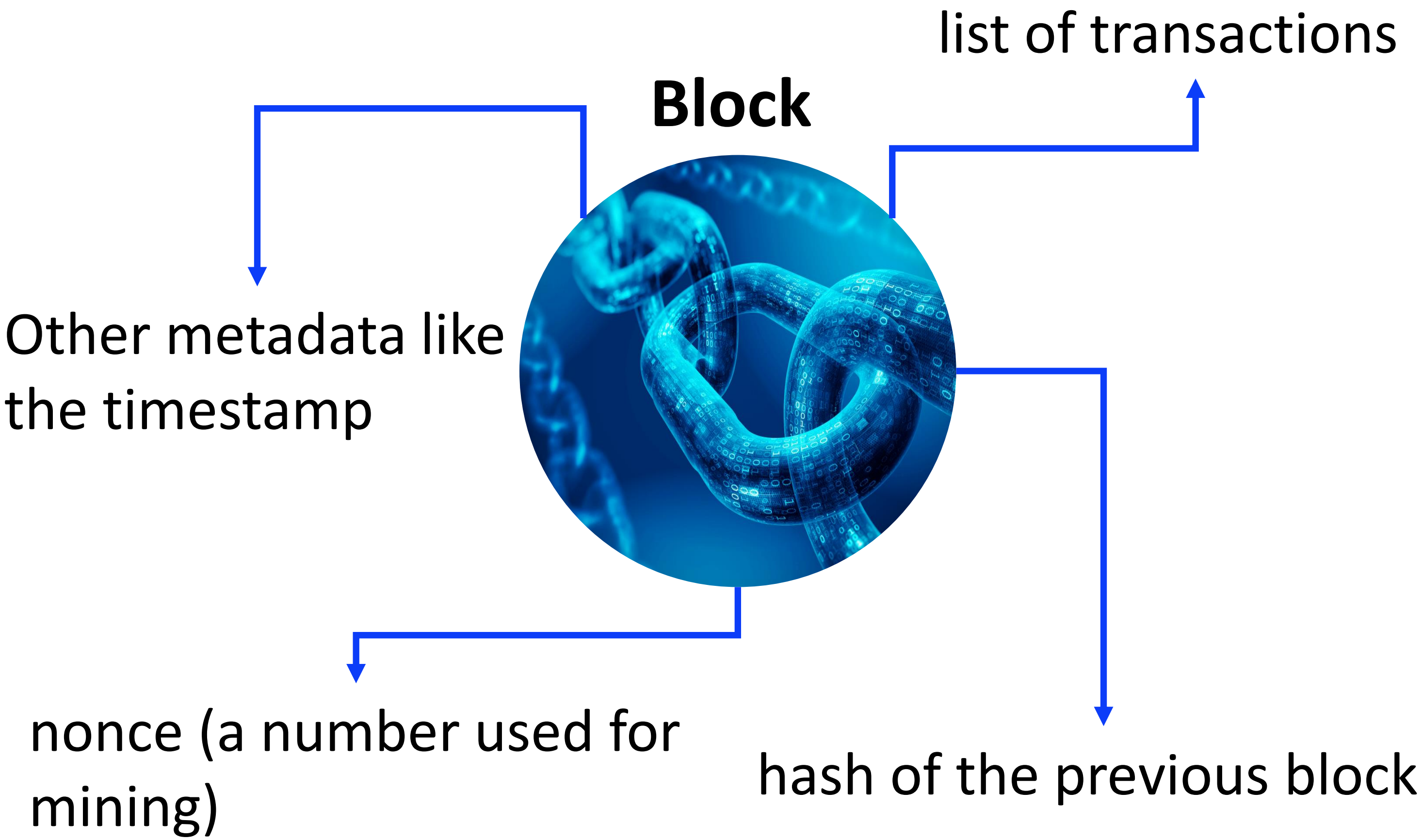
list of transactions

Block

nonce (a number used for mining)

hash of the previous block

Bitcoin


Block
list of transactions
Other metadata like the timestamp
nonce (a number used for mining)
hash of the previous block


DIGITAL * DECENTRALIZED * PEER TO PEER * CRYPTOCURRENCY * BITCOIN

Bitcoin

how does the blockchain work when Bob wants to send Bitcoin to Alice?

Bitcoin

Bob create transaction

Bitcoin

Bob create transaction

Bob wants to send 2 BTC to Alice.

Bitcoin

Bob create transaction

Bob wants to send 2 BTC to Alice.

1. **Sender Address:** Bob's Bitcoin address (derived from her public key).
2. **Recipient Address:** Alice's Bitcoin address.
3. **Amount:** 2 BTC.
4. **Digital Signature:** Bob uses his private key to sign the transaction, ensuring it’s authentic and can’t be altered.

Bitcoin

Broadcasting the transaction

Bitcoin

Broadcasting the transaction

The transaction is broadcast to the Bitcoin network (a peer-to-peer network of nodes)

Bitcoin

Broadcasting the transaction

The transaction is broadcast to the Bitcoin network (a peer-to-peer network of nodes)

- **Signature Validity:** Using Bob's public key, nodes verify that Bob signs the transaction.
- **Sufficient Balance:** Nodes confirm Bob has at least 2 BTC to spend.

Bitcoin

Broadcasting the transaction

The transaction is broadcast to the Bitcoin network (a peer-to-peer network of nodes)

- ✓ **Signature Validity:** Using Bob's public key, nodes verify that Bob signs the transaction.
- ✓ **Sufficient Balance:** Nodes confirm Bob has at least 2 BTC to spend.

If valid, the transaction is added to the mempool (a pool of pending transactions)

Bitcoin

Mining (Adding to the Blockchain)

Bitcoin

Mining (Adding to the Blockchain)

Miners group pending transactions into a new block. They compete to solve a Proof-of-Work (PoW) puzzle by finding a special number (the nonce) such that:

Hash(Block Data + Nonce) < Target Value

Bitcoin

Mining (Adding to the Blockchain)

Miners group pending transactions into a new block. They compete to solve a Proof-of-Work (PoW) puzzle by finding a special number (the nonce) such that:

Hash(Block Data + Nonce) < Target Value

Once a miner finds a solution:

1. The new block (containing Bob's transaction) is broadcast to the network
2. Other nodes verify the solution and the validity of the block

Bitcoin

Block added to Blockchain

If the block is valid, it is added to the blockchain. The blockchain now has a record of Bob sending 2 BTC to Alice

Bitcoin

Alice receives bitcoin

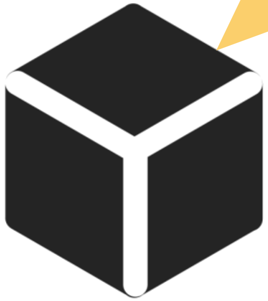
Bitcoin

Alice receives bitcoin

Alice's Bitcoin wallet checks the blockchain and sees a transaction crediting him with 2 BTC. The transaction is considered confirmed after multiple blocks are added after the block containing this transaction (this ensures security against potential forks)

Bitcoin

Block 1001
Transactions: [Charlie → Dana: 1 BTC]
Hash: H1001



Bitcoin

Block 1001
Transactions: [Charlie → Dana: 1 BTC]
Hash: H1001

Block 1002
Transactions: [Bob → Alice : 2 BTC]
The hash of Previous Block: H1001
Nonce: 982371
Hash: H1002

Bitcoin

Advantages and challenges

Bitcoin

Advantages and challenges

- Transparency
- security
- the ability to operate without intermediaries

- Scalability
- energy of mining
- regulatory concerns

Edge of science

Divisible E-Cash for Billing in Private Ad Retargeting

Kevin Liao
Henry Corrigan-Gibbs
Dan Boneh

2024

Link

Divisible E-Cash for Billing in Private Ad Retargeting

Kevin Liao
Henry Corrigan-Gibbs
Dan Boneh

2024

Link

Divisible E-Cash for Billing in Private Ad Retargeting

It examines a solution to solve the privacy problem in Ad Retargeting. The main goal of this research is to design a system that allows anonymous payment and maintains user privacy in the digital advertising process.

Kevin Liao
Henry Corrigan-Gibbs
Dan Boneh

2024

Link

Divisible E-Cash for Billing in Private Ad Retargeting

It examines a solution to solve the privacy problem in Ad Retargeting. The main goal of this research is to design a system that allows anonymous payment and maintains user privacy in the digital advertising process.

Specifically, the authors propose a system for divisible electronic cash (Divisible E-Cash) that allows users to make smaller transactions without revealing personal information or previous transactions. The system aims to achieve a balance between economic efficiency and privacy.

Kevin Liao
Henry Corrigan-Gibbs
Dan Boneh

2024

Link

Divisible E-Cash for Billing in Private Ad Retargeting

Zero-knowledge proofs

Kevin Liao
Henry Corrigan-Gibbs
Dan Boneh

2024

Link

Divisible E-Cash for Billing in Private Ad Retargeting

Zero-knowledge proofs

This system allows users to withdraw a large amount of currency and anonymously divide it into smaller portions.

Kevin Liao
Henry Corrigan-Gibbs
Dan Boneh

2024

Link

Divisible E-Cash for Billing in Private Ad Retargeting

Zero-knowledge proofs

This system allows users to withdraw a large amount of currency and anonymously divide it into smaller portions.

Anonymization protocol

Compatibility with advertising infrastructure

Kevin Liao
Henry Corrigan-Gibbs
Dan Boneh

2024

Link

Divisible E-Cash for Billing in Private Ad Retargeting

The system was evaluated based on simulation and real-world experiments, and the results showed that:

Kevin Liao
Henry Corrigan-Gibbs
Dan Boneh

2024

Link

Divisible E-Cash for Billing in Private Ad Retargeting

The system was evaluated based on simulation and real-world experiments, and the results showed that:

- The system latency is only about 63 milliseconds, which is quite acceptable for integration into online advertising.

Kevin Liao
Henry Corrigan-Gibbs
Dan Boneh

2024

Link

Divisible E-Cash for Billing in Private Ad Retargeting

The system was evaluated based on simulation and real-world experiments, and the results showed that:

- The system latency is only about 63 milliseconds, which is quite acceptable for integration into online advertising.
- The system performs better than previous methods in terms of anonymization and money segmentation.

Kevin Liao
Henry Corrigan-Gibbs
Dan Boneh

2024

Link

Divisible E-Cash for Billing in Private Ad Retargeting

The system was evaluated based on simulation and real-world experiments, and the results showed that:

- The system latency is only about 63 milliseconds, which is quite acceptable for integration into online advertising.
- The system performs better than previous methods in terms of anonymization and money segmentation.
- The scalability and security of the system are guaranteed, and it can be used on a large scale for advertising networks.

HybCBDC: A Design for Central Bank Digital Currency(CBDC) Systems Enabling Digital Cash

HybCBDC: A Design for Central Bank Digital Currency(CBDC) Systems Enabling Digital Cash

RICKY LAMBERTY
DANIEL KIRSTE
NICLAS KANNENGIEBER
ALI SUNYAEV

2 October 2024

Link

RICKY LAMBERTY
DANIEL KIRSTE
NICLAS KANNENGIEBER
ALI SUNYAEV

2 October 2024

Link

HybCBDC: A Design for Central Bank Digital Currency(CBDC) Systems Enabling Digital Cash

The main purpose of CBDC is to support retail and wholesale.

RICKY LAMBERTY
DANIEL KIRSTE
NICLAS KANNENGIEBER
ALI SUNYAEV

2 October 2024

Link

HybCBDC: A Design for Central Bank Digital Currency(CBDC) Systems Enabling Digital Cash

The main purpose of CBDC is to support retail and wholesale.

Transparency

RICKY LAMBERTY
DANIEL KIRSTE
NICLAS KANNENGIEBER
ALI SUNYAEV

2 October 2024

Link

HybCBDC: A Design for Central Bank Digital Currency(CBDC) Systems Enabling Digital Cash

The main purpose of CBDC is to support retail and wholesale.

Transparency

Transparency in this system conflicts with these rules.

- Anti-money laundering(AML)
- Countering the finance of terrorism(CFT)

RICKY LAMBERTY
DANIEL KIRSTE
NICLAS KANNENGIEBER
ALI SUNYAEV

2 October 2024

Link

HybCBDC: A Design for Central Bank Digital Currency(CBDC) Systems Enabling Digital Cash

HybCBDC can solve this problem!

RICKY LAMBERTY
DANIEL KIRSTE
NICLAS KANNENGIEBER
ALI SUNYAEV

2 October 2024

Link

HybCBDC: A Design for Central Bank Digital Currency(CBDC) Systems Enabling Digital Cash

HybCBDC can solve this problem!

Layer privacy Model:

RICKY LAMBERTY
DANIEL KIRSTE
NICLAS KANNENGIEBER
ALI SUNYAEV

2 October 2024

Link

HybCBDC: A Design for Central Bank Digital Currency(CBDC) Systems Enabling Digital Cash

HybCBDC can solve this problem!

Layer privacy Model:

$$\text{Transactions}\begin{cases}\text{Low-risk}\\ \text{High-risk}\end{cases}$$

RICKY LAMBERTY
DANIEL KIRSTE
NICLAS KANNENGIEBER
ALI SUNYAEV

2 October 2024

Link

HybCBDC: A Design for Central Bank Digital Currency(CBDC) Systems Enabling Digital Cash

HybCBDC can solve this problem!

Layer privacy Model:

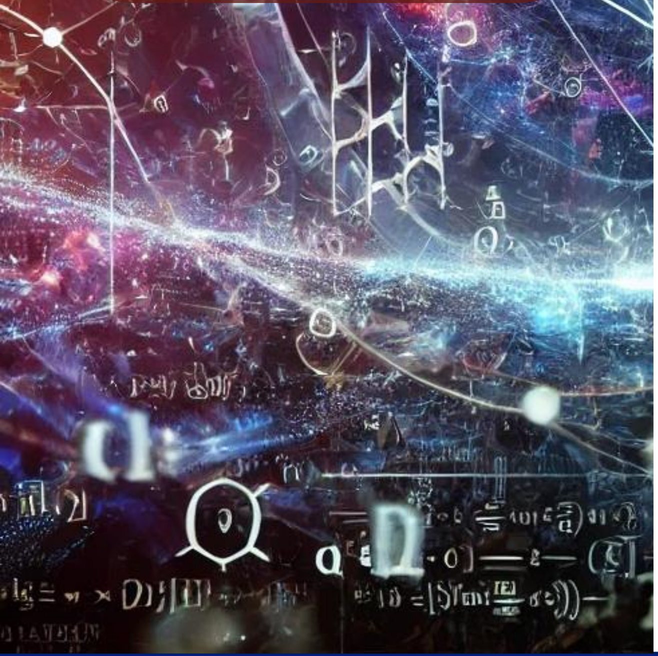
Transactions
- Low-risk
- High-risk

Only specific entities have access to user’s data!

RICKY LAMBERTY
DANIEL KIRSTE
NICLAS KANNENGIEBER
ALI SUNYAEV

2 October 2024

Link



HybCBDC: A Design for Central Bank Digital Currency(CBDC) Systems Enabling Digital Cash

HybCBDC can solve this problem!

Layer privacy Model:

Transactions $\begin{cases} \text{Low-risk} \\ \text{High-risk} \end{cases}$

Only specific entities have access to user’s data!

Role of intermediaries: central banks don’t store sensitive data and some regulated intermediaries check transactions.

RICKY LAMBERTY
DANIEL KIRSTE
NICLAS KANNENGIEBER
ALI SUNYAEV

2 October 2024

Link

HybCBDC: A Design for Central Bank Digital Currency(CBDC) Systems Enabling Digital Cash

Limits:

1. HybCBDC increases Complexity and operational costs.
2. It doesn’t support offline payments.
3. In implementing protocols, it is assumed that the attacker does not have access to transaction information (such as IP, etc.), while in the real world, this assumption is not true.

Thank You

sajjadranjbaryazdi@gmail.com