# Blockchain-based Transcript Inquiry and Verification System

**Author**: ZHIHAO,LIANG **Supervisor**: Bertrand M.T. Lin

梁志豪 林妙聰

National Chiao Tung University
Department of Information Management and Finance

TANET 2017

# Outline:

Outline
Motivation and Problem Statement
Background
System Architecture
Results and discussion
References

# Motivation and Problem Statement:

Outline

Motivation and Problem Statement

Background

System Architecture

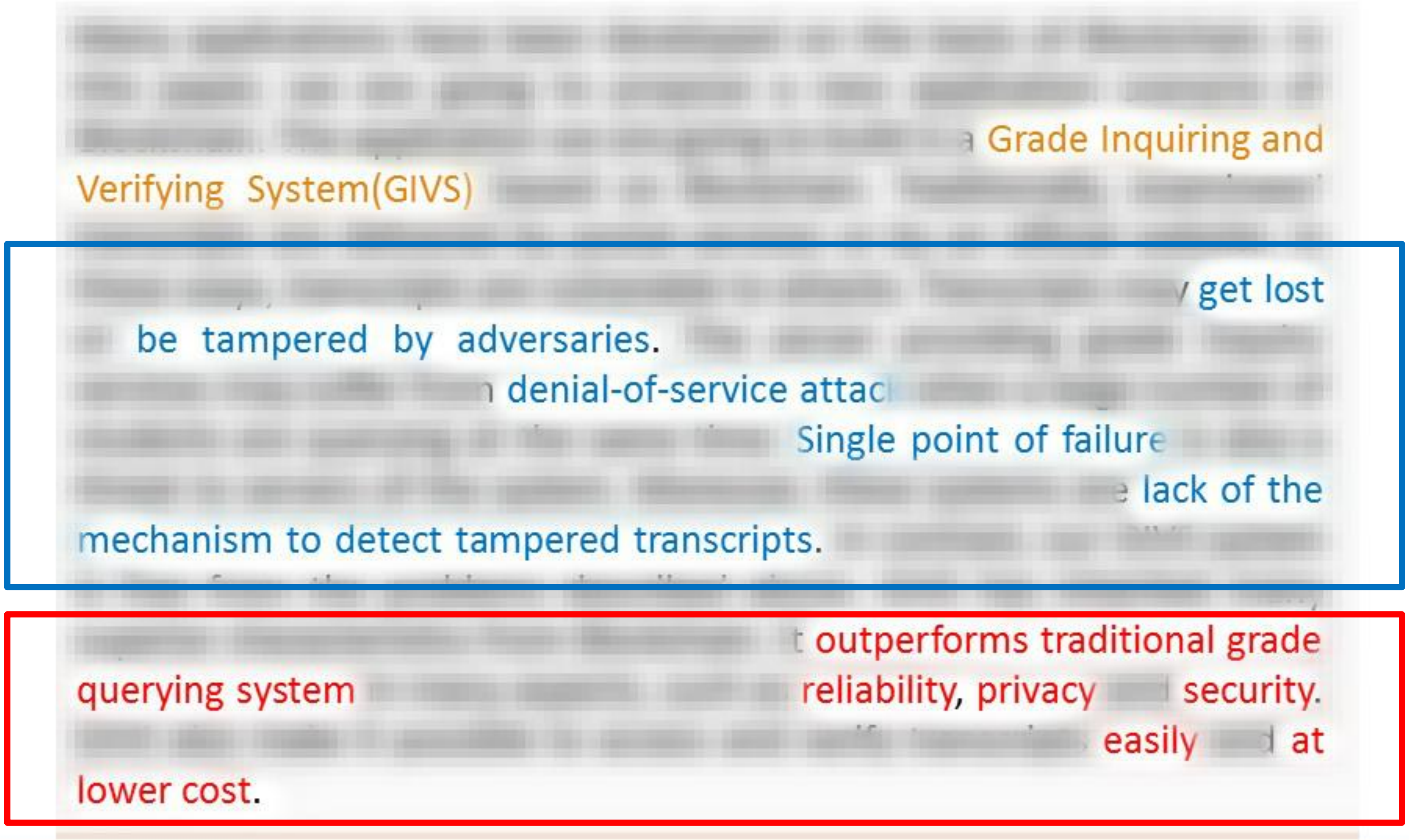
Results and discussion

References

# Motivation and Problem Statement:

Many applications have been developed on the basis of Blockchain. In this paper, we are going to propose a new application scenario of Blockchain. The application we are going to build is a Grade Inquiry and Verification System(GIVS) based on Blockchain. Traditionally, examinees' transcripts are delivered by postal services or by an official website. In these ways, transcripts are vulnerable to attacks. Transcripts may get lost or be tampered by adversaries. The server providing grade inquiry services may suffer from denial-of-service attack when a large number of students are querying at the same time. Single point of failure is also a threat to servers of the system. Moreover, these systems are lack of the mechanism to detect tampered transcripts. In contrast, our GIVS system is free from the problems described above. GIVS has inherited many superior characteristics from Blockchain. It outperforms traditional grade querying systems in many aspects, such as reliability, privacy and security. GIVS also make it possible to access and verify transcripts easily and at lower cost.

# Motivation and Problem Statement:



Grade Inquiring and Verifying System(GIVS)

Traditional

get lost

be tampered by adversaries.

denial-of-service attac

Single point of failure

lack of the mechanism to detect tampered transcripts.

GIVS

outperforms traditional grade querying system

reliability, privacy

security.

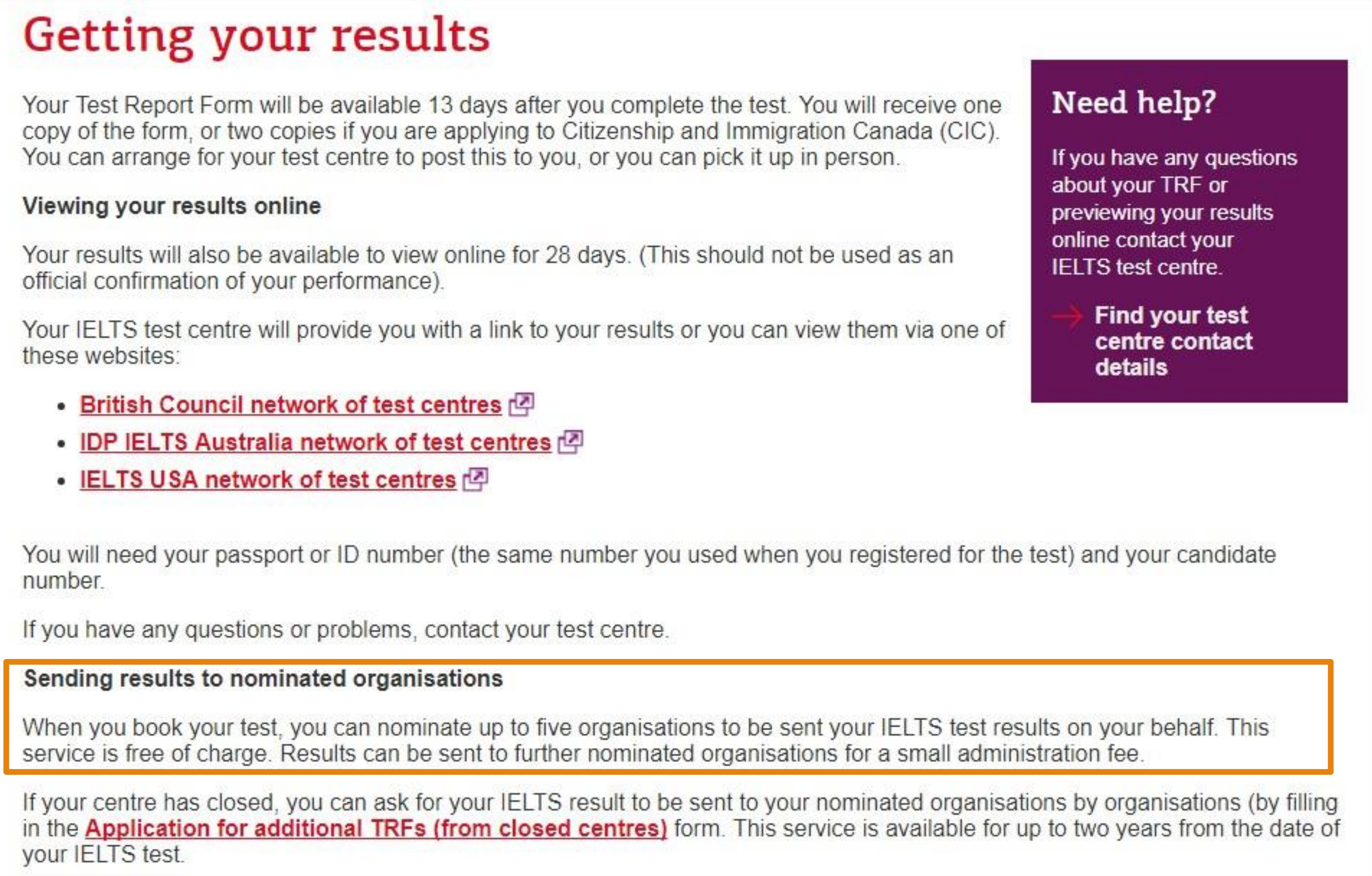
easily

at lower cost.

A TRANSCRIPT QUERY AND VERIFYING SYSTEM BASED ON BLOCKCHAIN

# IELTS:

- Administration fee
- Slow
- Complex
- May be tampered



## Getting your results

Your Test Report Form will be available 13 days after you complete the test. You will receive one copy of the form, or two copies if you are applying to Citizenship and Immigration Canada (CIC). You can arrange for your test centre to post this to you, or you can pick it up in person.

**Viewing your results online**

Your results will also be available to view online for 28 days. (This should not be used as an official confirmation of your performance).

Your IELTS test centre will provide you with a link to your results or you can view them via one of these websites:

- British Council network of test centres
- IDP IELTS Australia network of test centres
- IELTS USA network of test centres

You will need your passport or ID number (the same number you used when you registered for the test) and your candidate number.

If you have any questions or problems, contact your test centre.

**Sending results to nominated organisations**

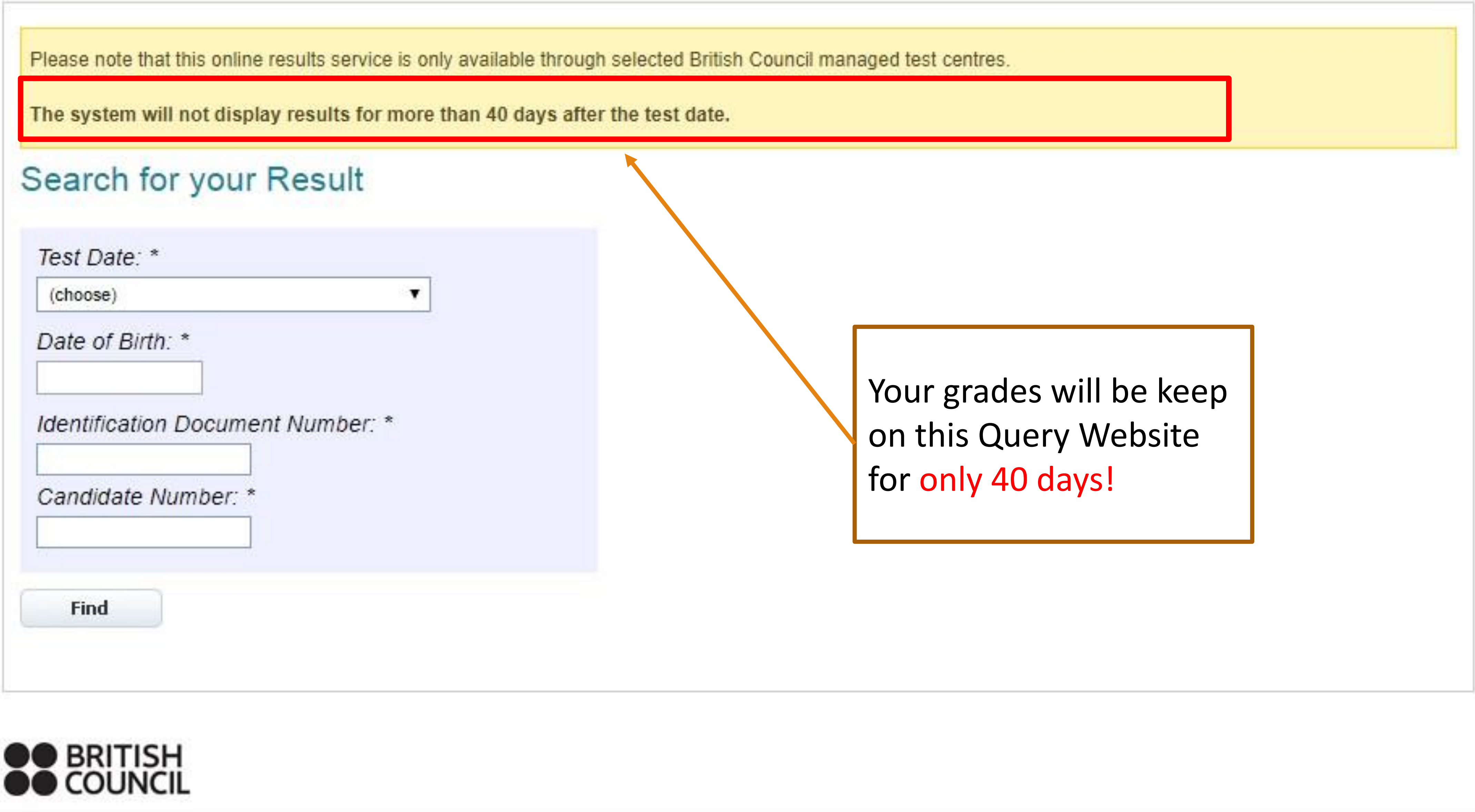
When you book your test, you can nominate up to five organisations to be sent your IELTS test results on your behalf. This service is free of charge. Results can be sent to further nominated organisations for a small administration fee.

If your centre has closed, you can ask for your IELTS result to be sent to your nominated organisations by organisations (by filling in the Application for additional TRFs (from closed centres) form. This service is available for up to two years from the date of your IELTS test.

### Need help?

If you have any questions about your TRF or previewing your results online contact your IELTS test centre.

→ **Find your test centre contact details**



Please note that this online results service is only available through selected British Council managed test centres.

**The system will not display results for more than 40 days after the test date.**

## Search for your Result

*Test Date:* *

(choose) ▼

*Date of Birth:* *

*Identification Document Number:* *

*Candidate Number:* *

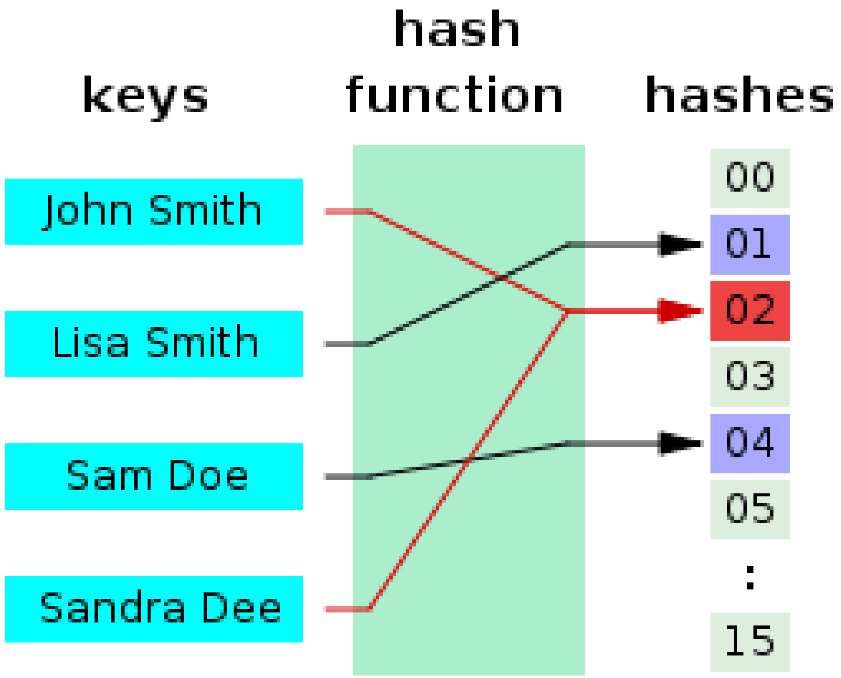
Your grades will be keep on this Query Website for only 40 days!

# Background:

Abstract
Outline
Motivation and Problem Statement
Background
System Architecture
Results and discussion
References

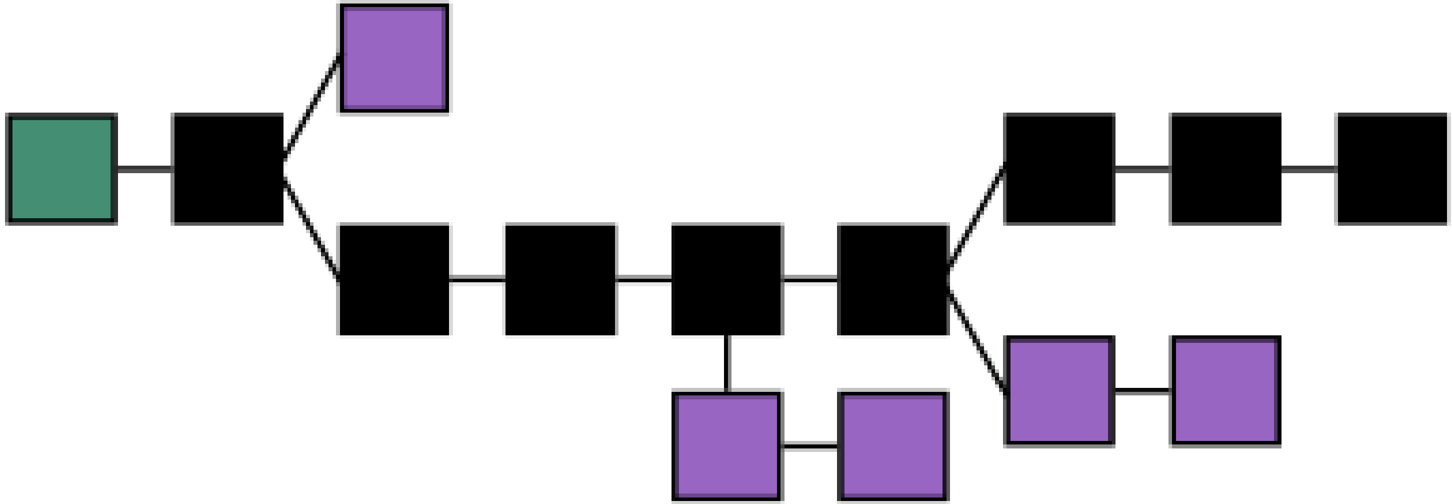
I. Hash function
II. Blockchain
III. Bitcoin Null Data Transaction
IV. Hierarchical deterministic wallet

# Hash function:



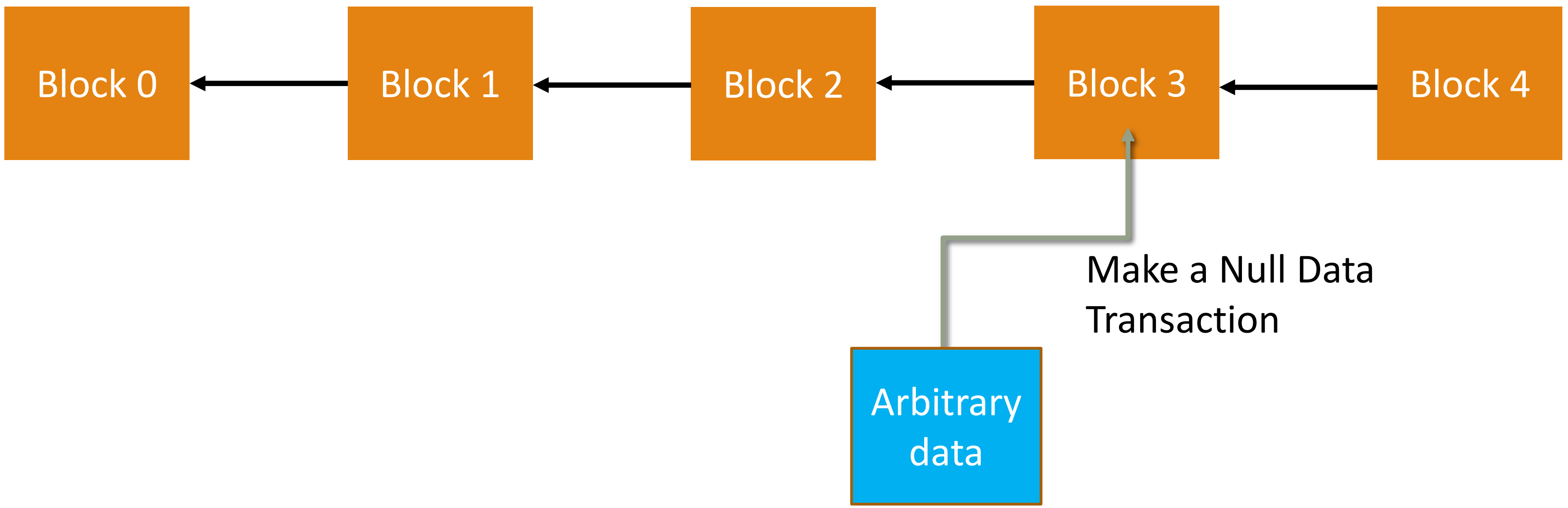
A hash function is any function that can be used to map data of arbitrary size to data of fixed size.

## Blockchain:



A blockchain is a continuously growing list of records, called blocks, which are linked and secured using cryptography. Each block typically contains a hash pointer as a link to a previous block, a timestamp and transaction data. By design, blockchains are inherently resistant to modification of the data.

# Bitcoin Null Data Transaction:

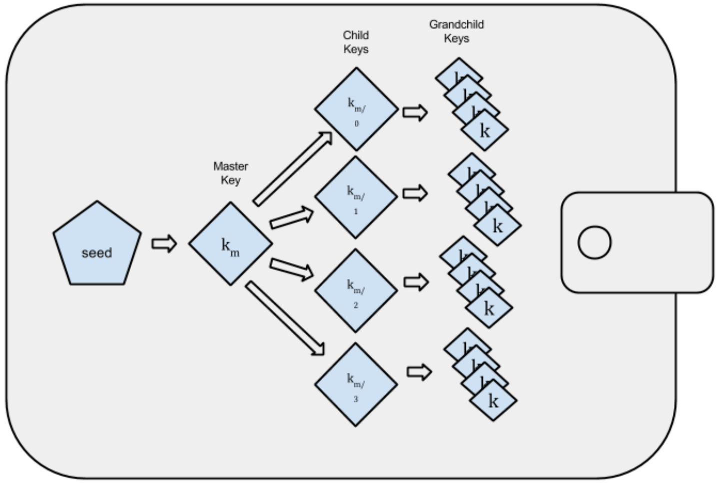


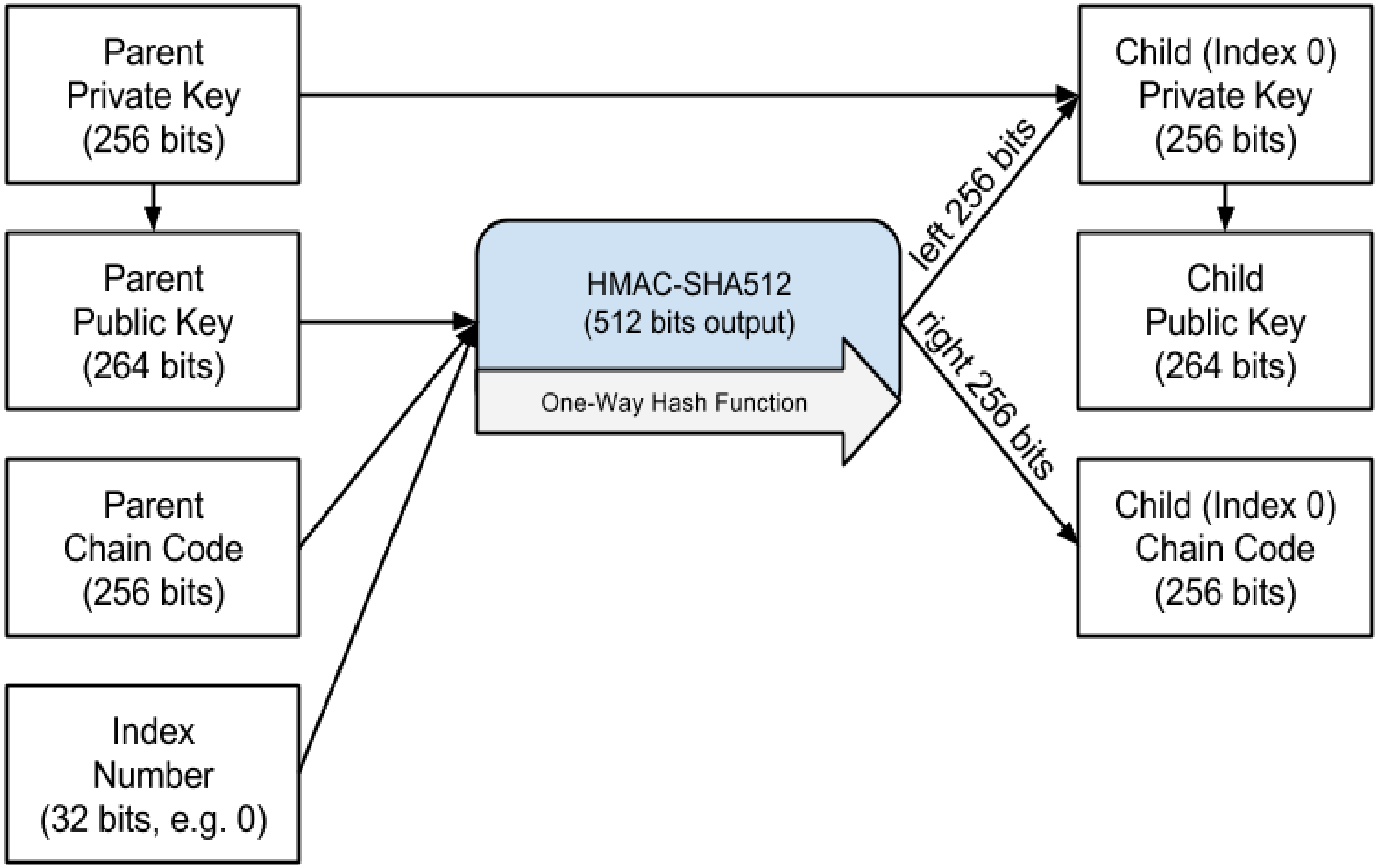
**Null data transaction** type relayed and mined by default in Bitcoin that adds arbitrary data to a provably unspendable pubkey script.

# Hierarchical deterministic wallet:

## Keys:

- Private key
- Public Key

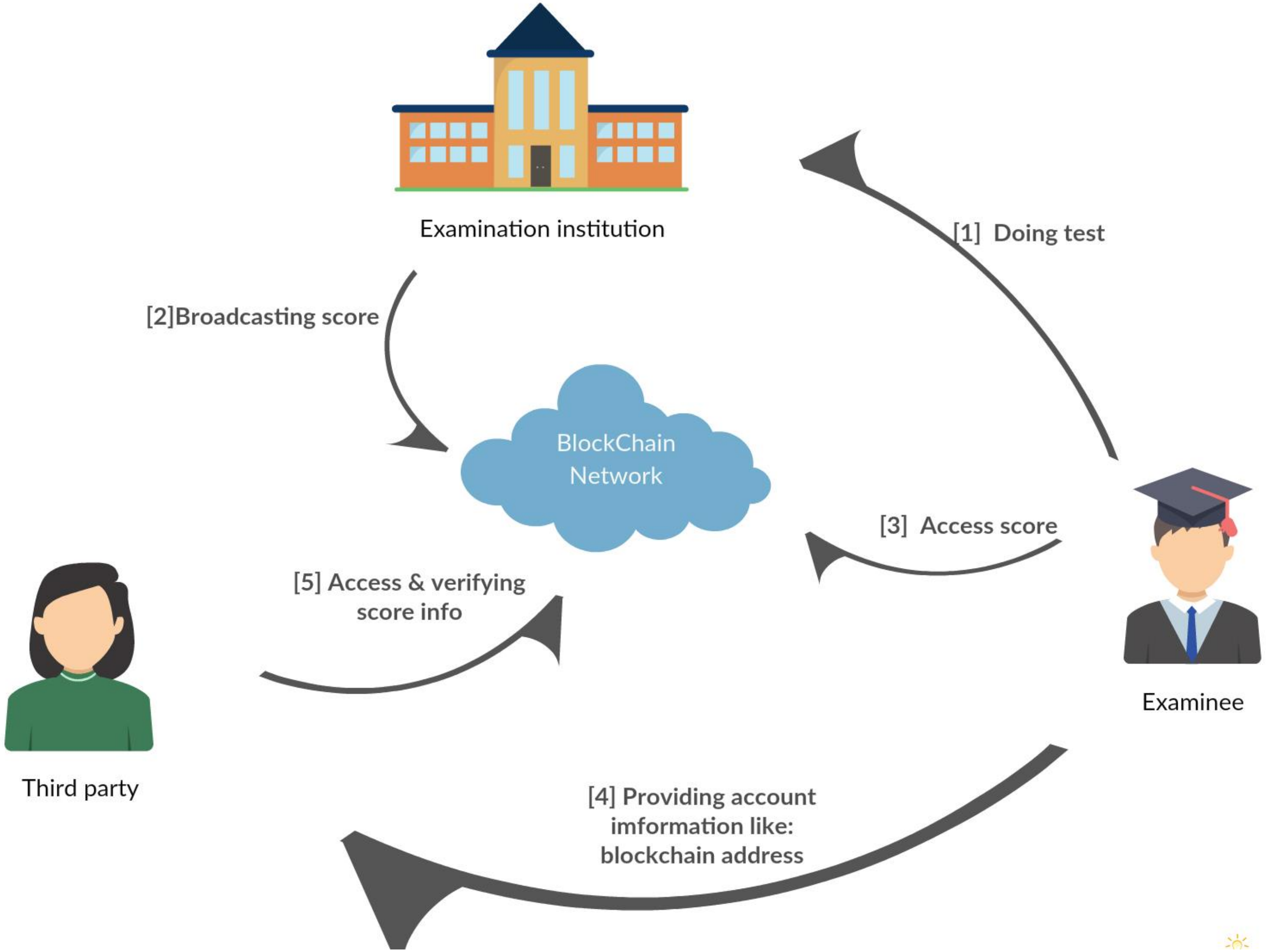



Parent Private Key (256 bits)
Parent Public Key (264 bits)
Parent Chain Code (256 bits)
Index Number (32 bits, e.g. 0)
HMAC-SHA512 (512 bits output)
One-Way Hash Function
left 256 bits
right 256 bits
Child (Index 0) Private Key (256 bits)
Child Public Key (264 bits)
Child (Index 0) Chain Code (256 bits)

# System Architecture:

Outline
Motivation and Problem Statement
Background
System Architecture
Results and discussion
References

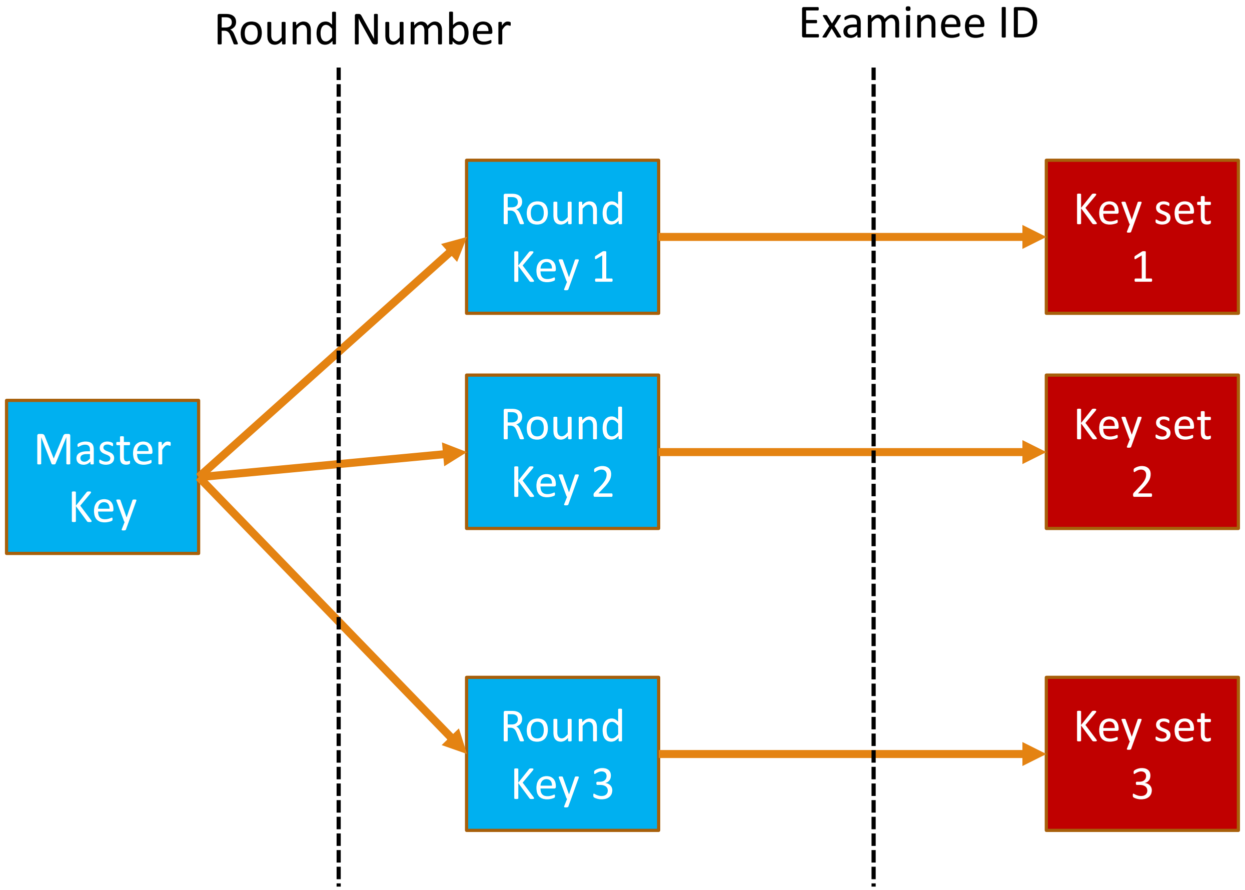
# Operating Procedures of GIVS system:



# Account Management :

**Examinees will receive:**

- Round Number
- Examinee ID
- His/Her Keys( private key and public key)

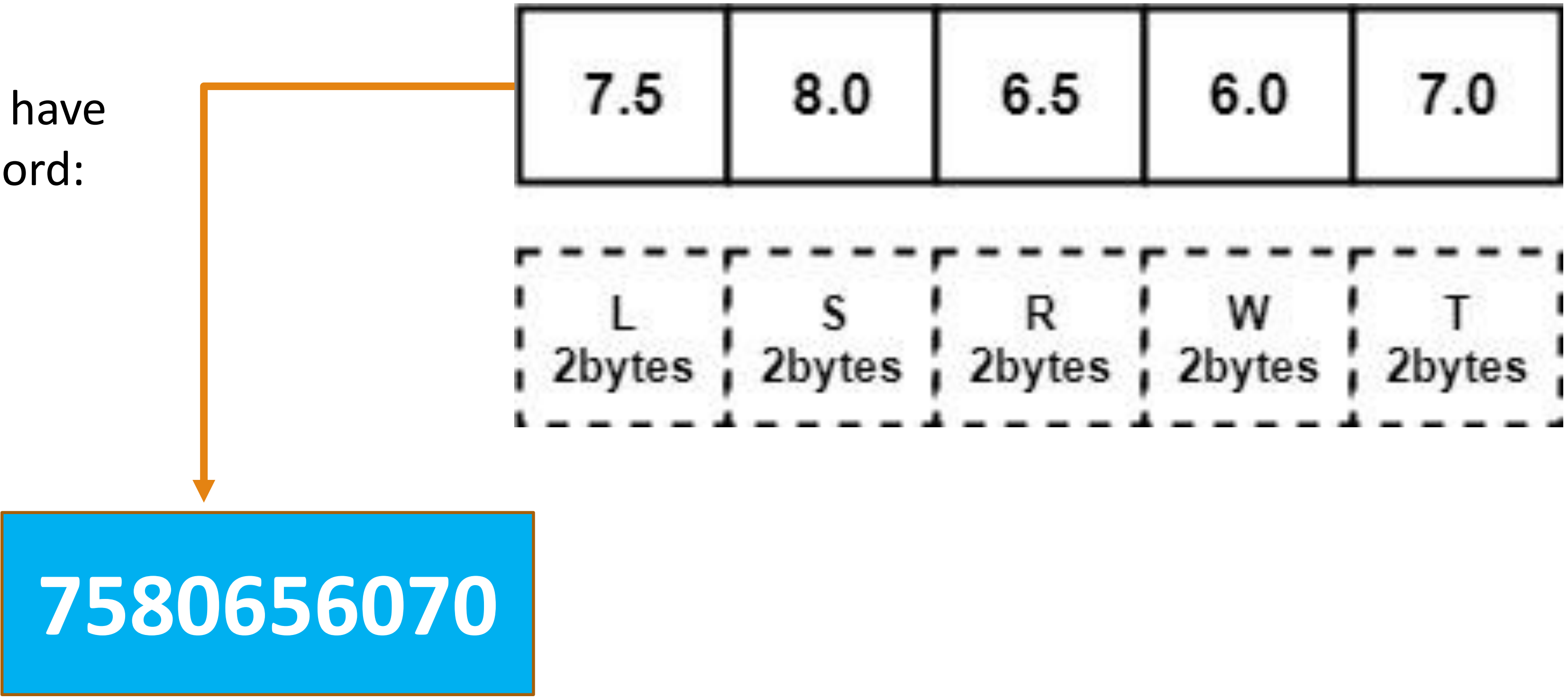


**Master public key is known to all!**

# Transcript format:

Take IELTS as an example. We have 5 pieces of information to record:

- Listening
- Speaking
- Reading
- Writing
- Total Grade



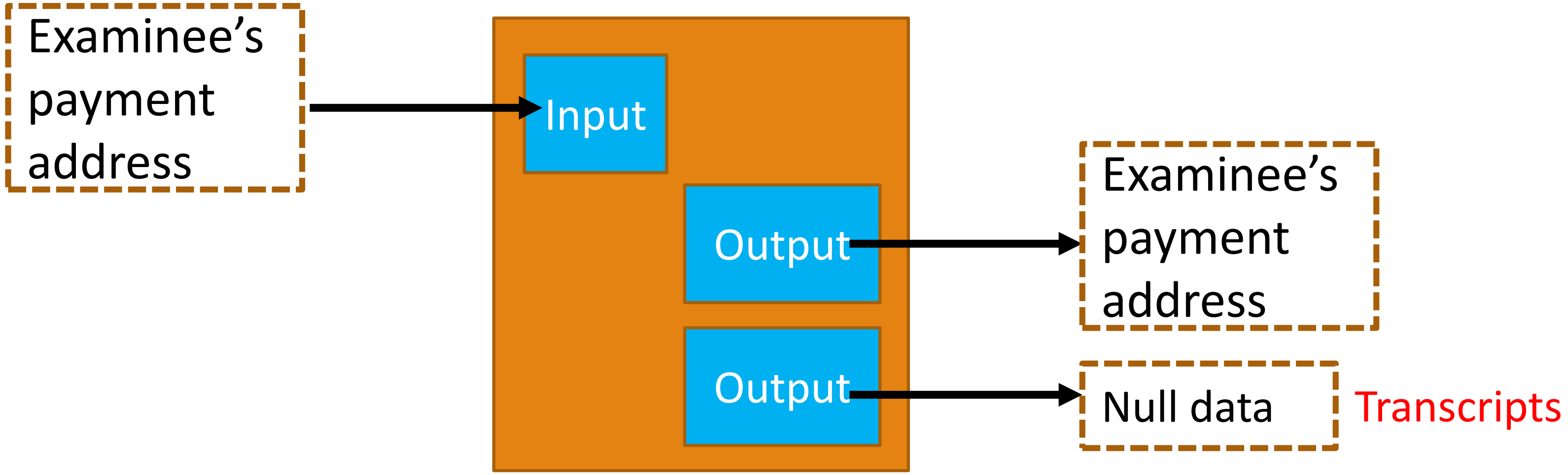
| 7.5 | 8.0 | 6.5 | 6.0 | 7.0 |
|---|---|---|---|---|
| L<br>2bytes | S<br>2bytes | R<br>2bytes | W<br>2bytes | T<br>2bytes |

7580656070

**Simple transcript**

# Transcripts Broadcasting:

1) **Each examinee** needs to prepare enough of transaction fees for his/her own payment address of blockchain.
2) **Examination Institution** builds transactions with transcripts attached to them, and broadcast them to blockchain.
3) **The input** and **output** address are examinee's payment address



**Transaction**

## Security issue:

Examinees may make similar transactions themselves to fabricate new transcripts for their interest.

## Solution:

- Pay to script hash
- 2 of 3 Multi-signature

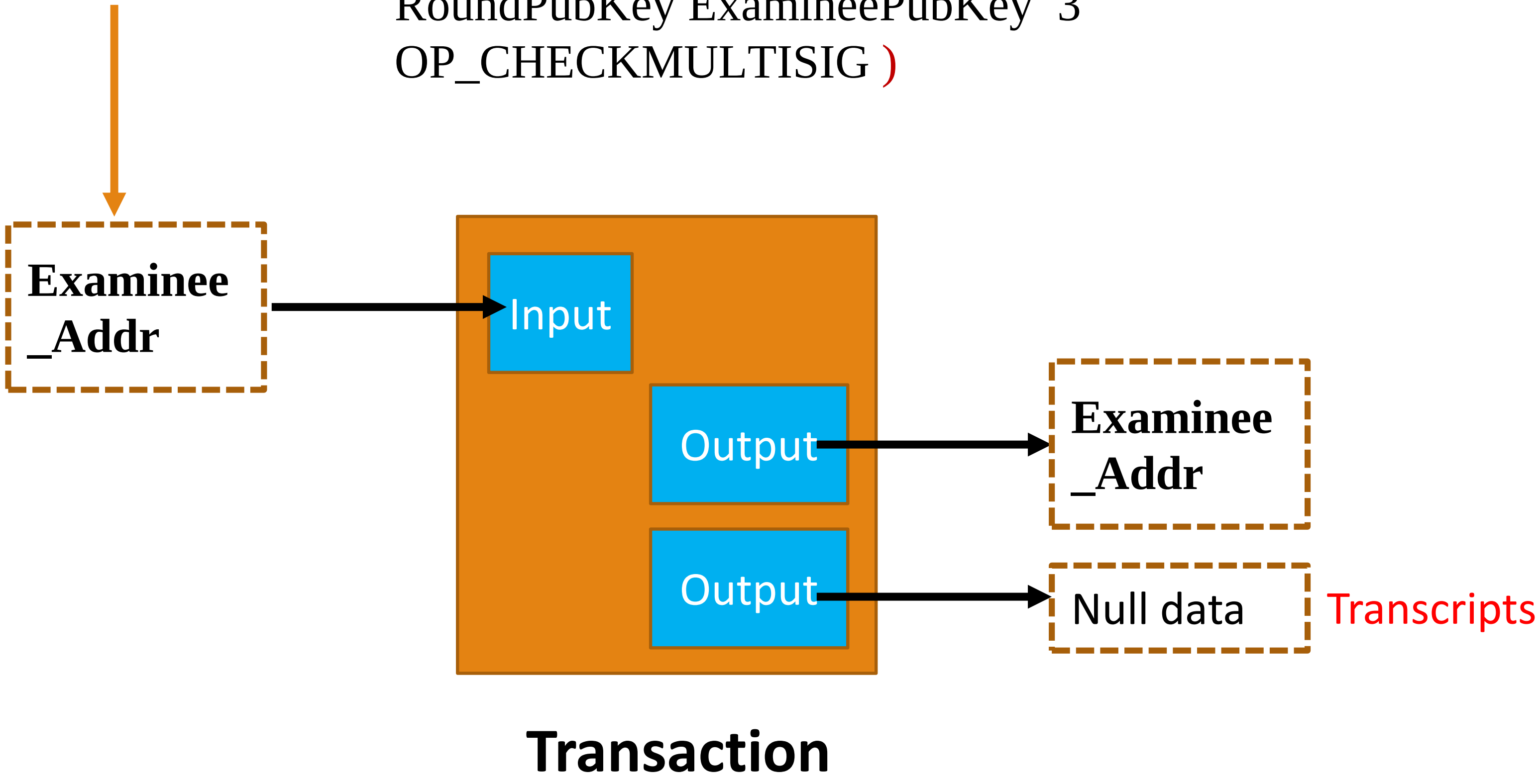
Only the one who holds at least 2 of Master Private Key, Round Private Key and Examinee Private Key, can make the transaction.

The real address of the examinee is:

**Examinee_Addr** = RIPEMD160( 2 MasterPubKey
RoundPubKey ExamineePubKey 3
OP_CHECKMULTISIG )

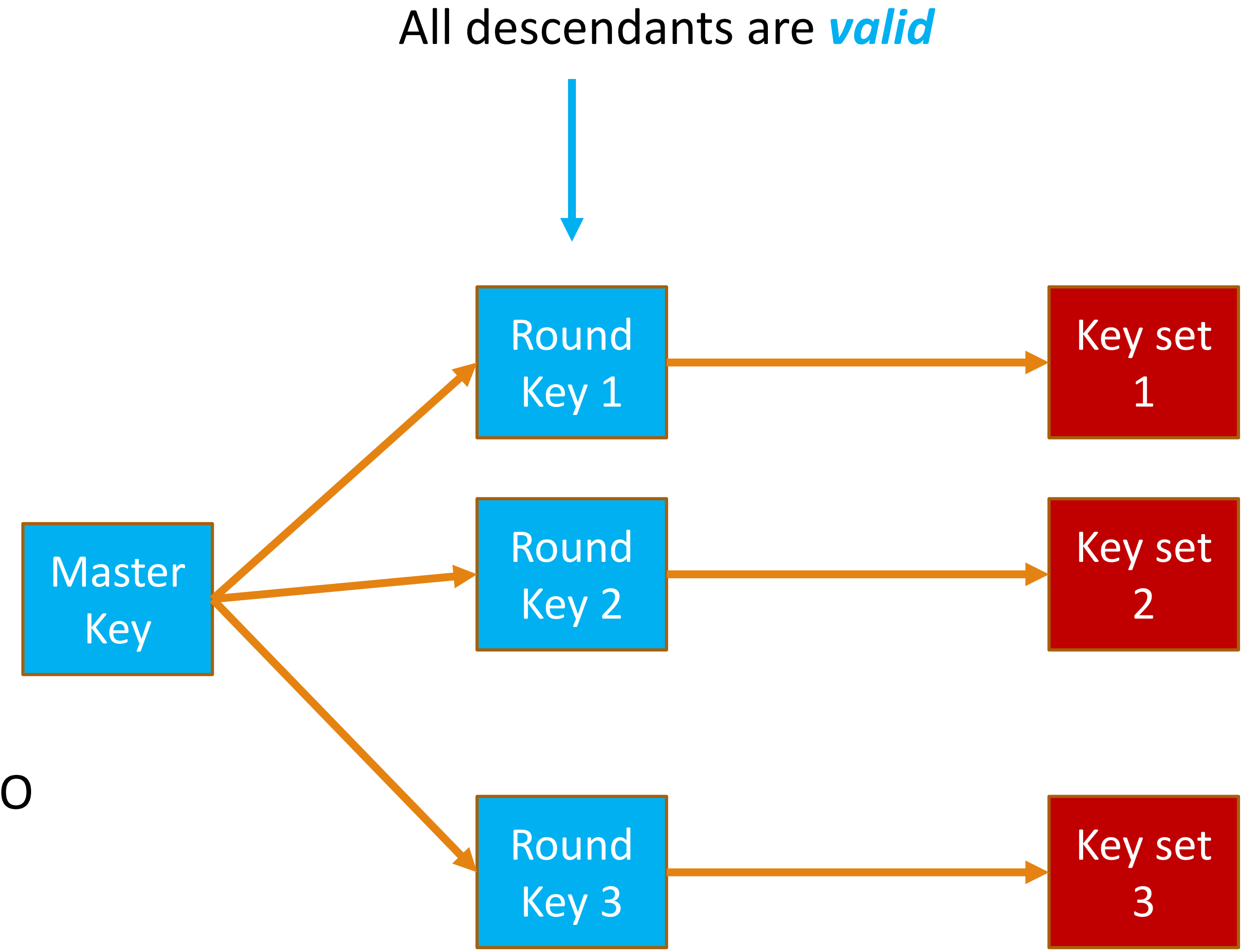
# Revised transaction:

**Examinee_Addr** = RIPEMD160( 2 MasterPubKey RoundPubKey ExamineePubKey 3 OP_CHECKMULTISIG )



# Revocation:

Transcripts have to be revoked if they are expired.



## Mechanism:

**Valid**: Round Key Address has UXTO
**Expired**: Round Key Address has no UXTO

**Note:** **UXTO**: Unspent Transaction Output

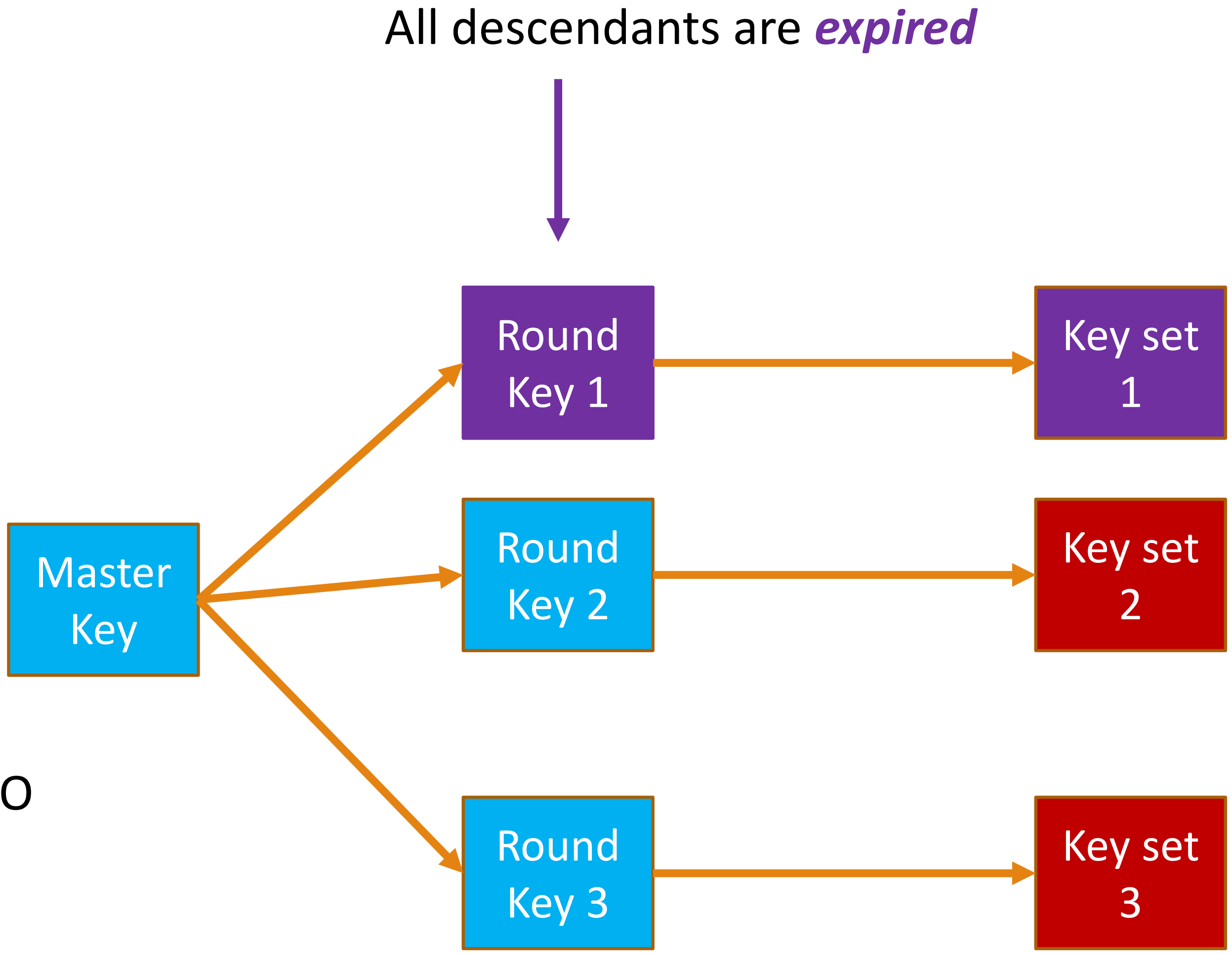
# Revocation(.cont) :

Transcripts have to be revoked if they are expired.

## Mechanism:

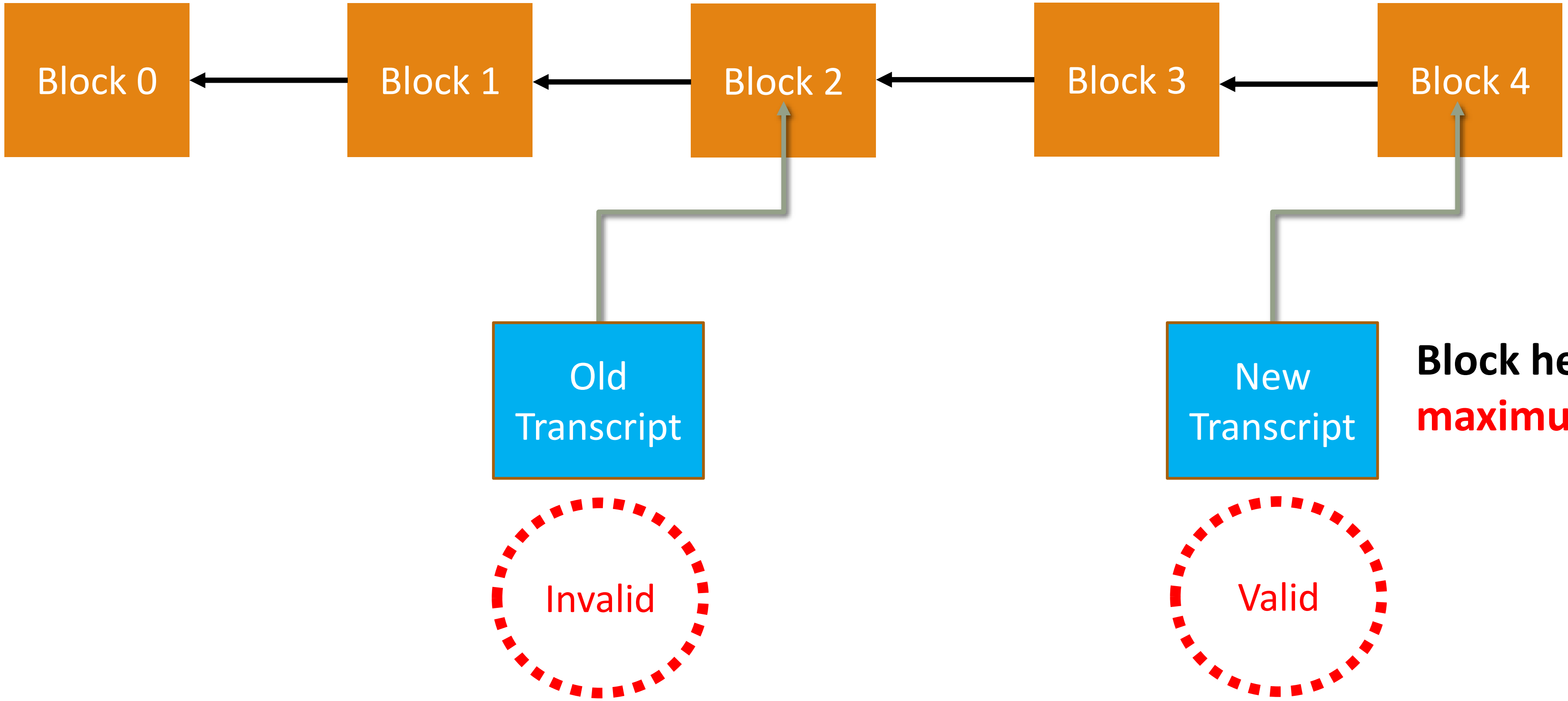
**Valid**: Round Key Address has UXTO

**Expired** : Round Key Address has no UXTO



**Note:** **UXTO**: Unspent Transaction Output

# Enquiry on Results:



Block height is **maximum!**

# Privacy:

To protect the privacy of examinees, examination institution need to encrypt transcripts before broadcasting them to network.

Encryption:

$C = AES256\_encrypt( M, hash256(ExamineePriKey) )$

*Decryption:*

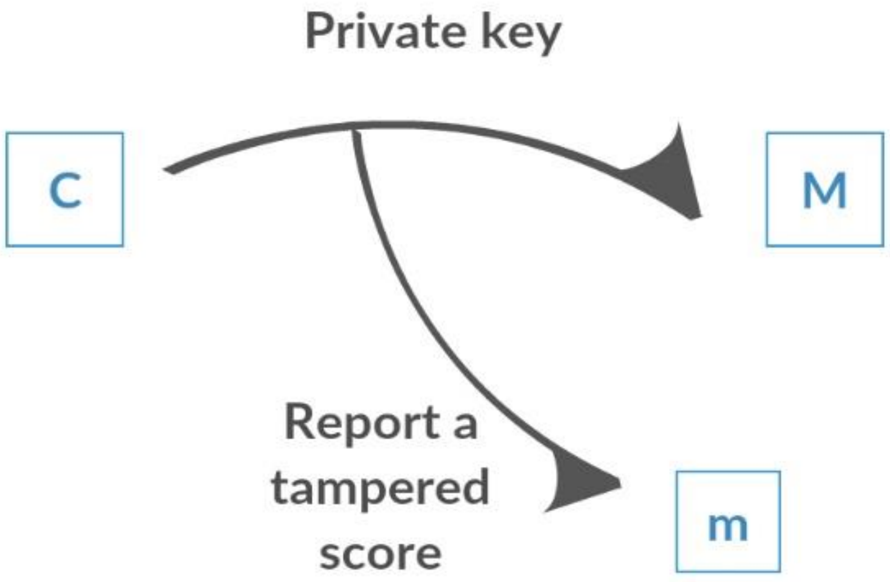
$M = AES256\_decrypt( C, hash256( ExamineePriKey ))$

**However, this will lead to another problem:**

# Privacy(.cont 1):

When the third party has obtained **C** from **Examinee_Addr,** it's hard to get **M** without knowing the private key of the examinee.

If the third party invite the examinee to decrypt the cyphertext **C**, he is unable to be sure that the examinee is honest.



# Privacy(.cont 2):

## Solution:

Attach hash of the transcript to the null data transaction before sending it to the network.

Denote $T$ as a random number of a reasonable size.

Then, the message of Null Data transaction is:

$$Null\_data\_Message = OP\_RETURN\!< C \,||\, hash160(M \,||\, T) \;>$$

Note:

Encryption:

$C = AES256\_encrypt(\, M||T,\, hash256(ExamineePriKey)\,)$

*Decryption:*

$M||T = AES256\_decrypt(\, C,\, hash256(ExamineePriKey)\,)$

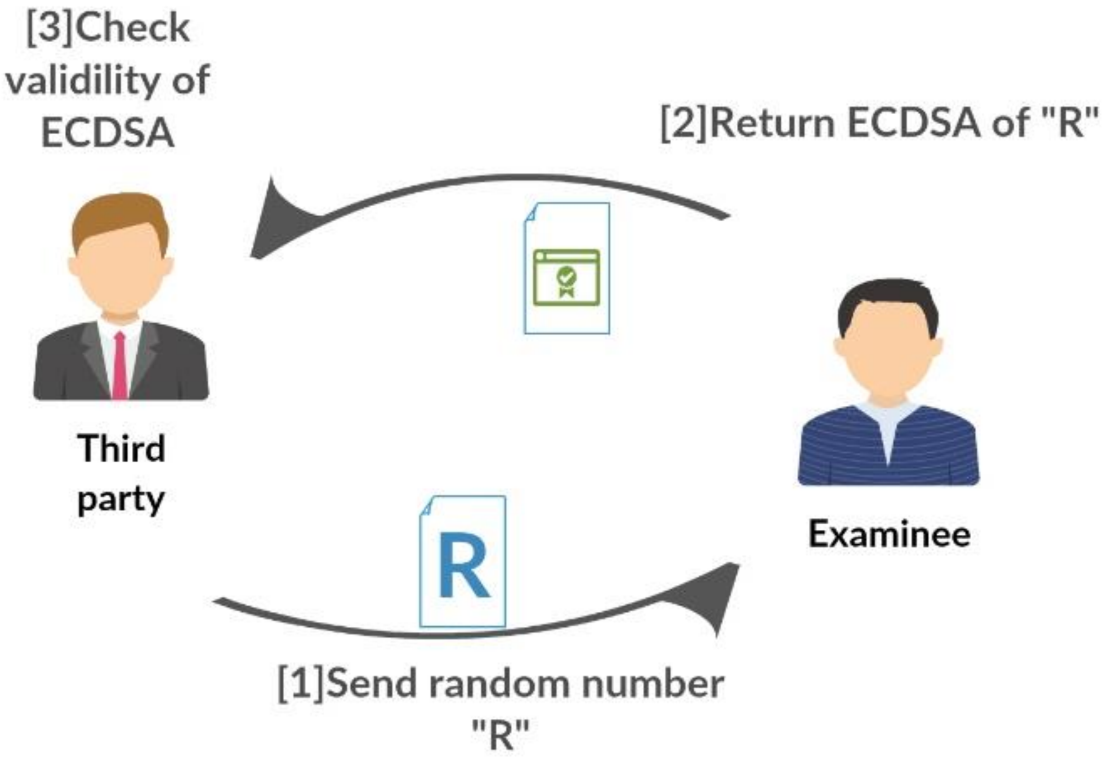
**Hash160()** must be a hash function with hiding property.

# Authentication:

**[The first step]**

The third party receives the public key from the examinee. And verify that it’s a descendant of Master public key.

**[The second step]**



# Verifying transcript:

The third party invites examinee to decrypt **C**, and receive **M’||T’** from him.
Then the third party does the following to check the validity of the reported transcript **M’**

$$\boxed{Validity = is\_equal(\ hash160(M'||T'),\ hash160(M||T)\ )}$$

If validity is true, then we know that **M = M’**,and the transcript is valid.

# Results:

Outline

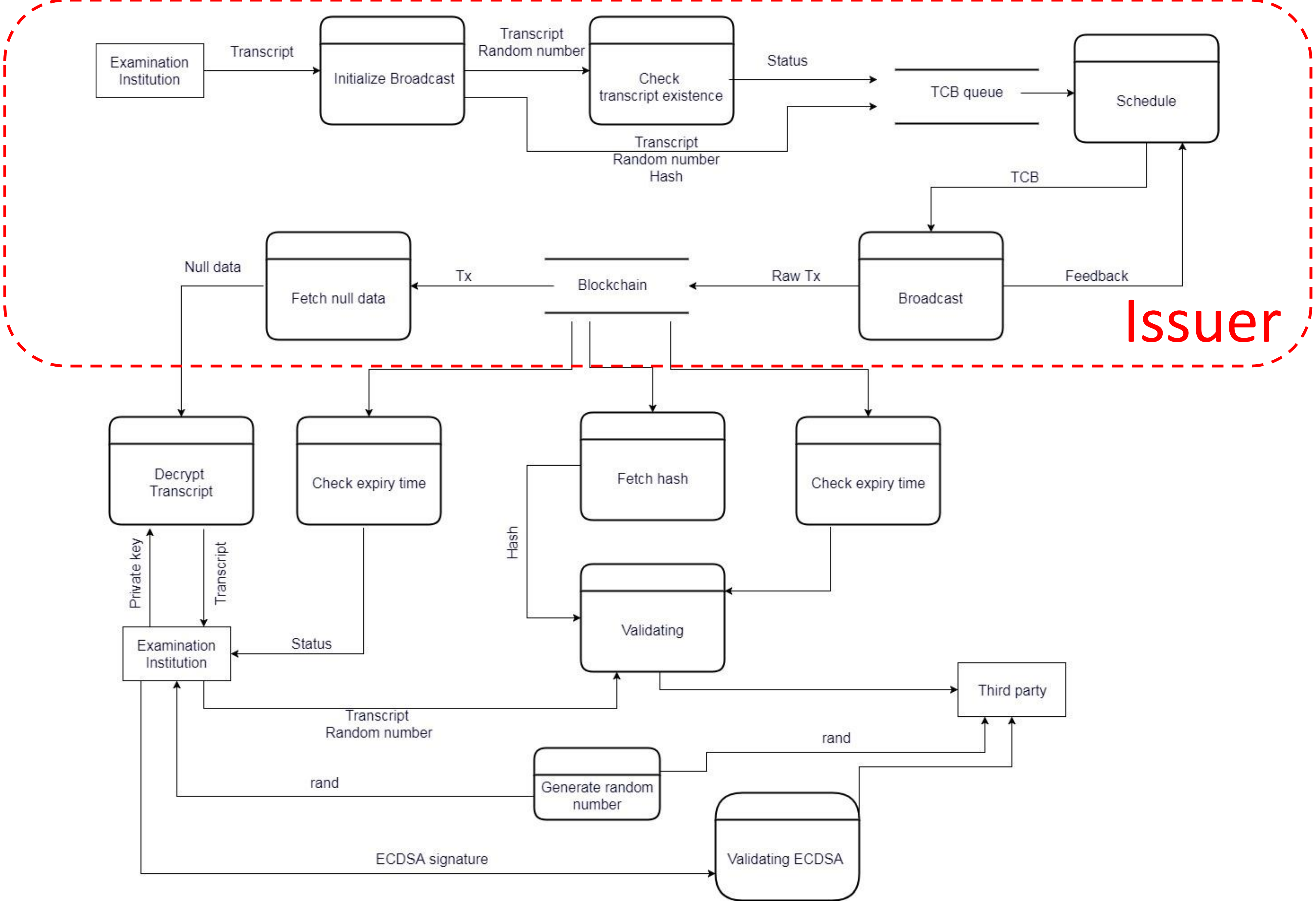
Motivation and Problem Statement

Background

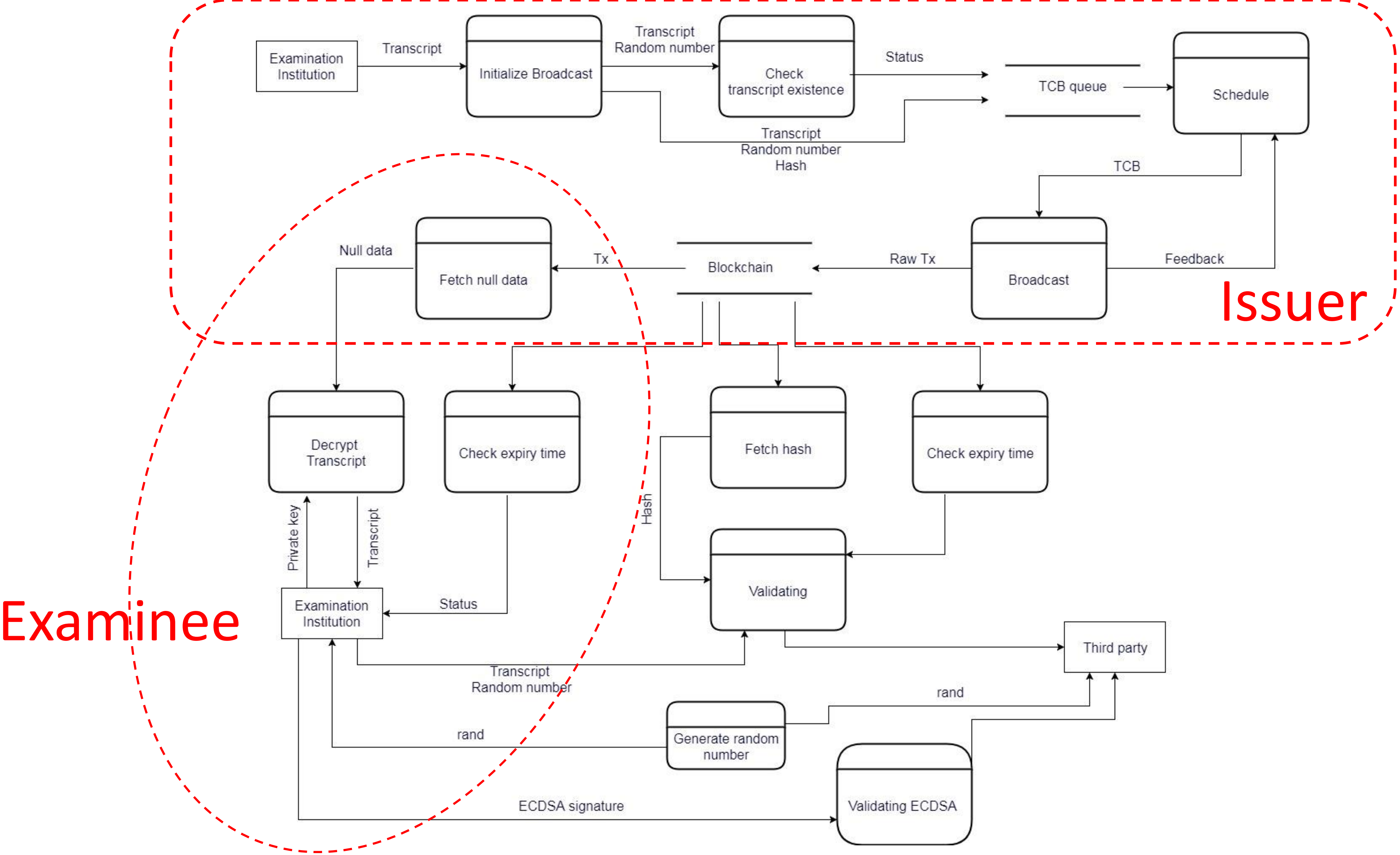
System Architecture

Results and discussion

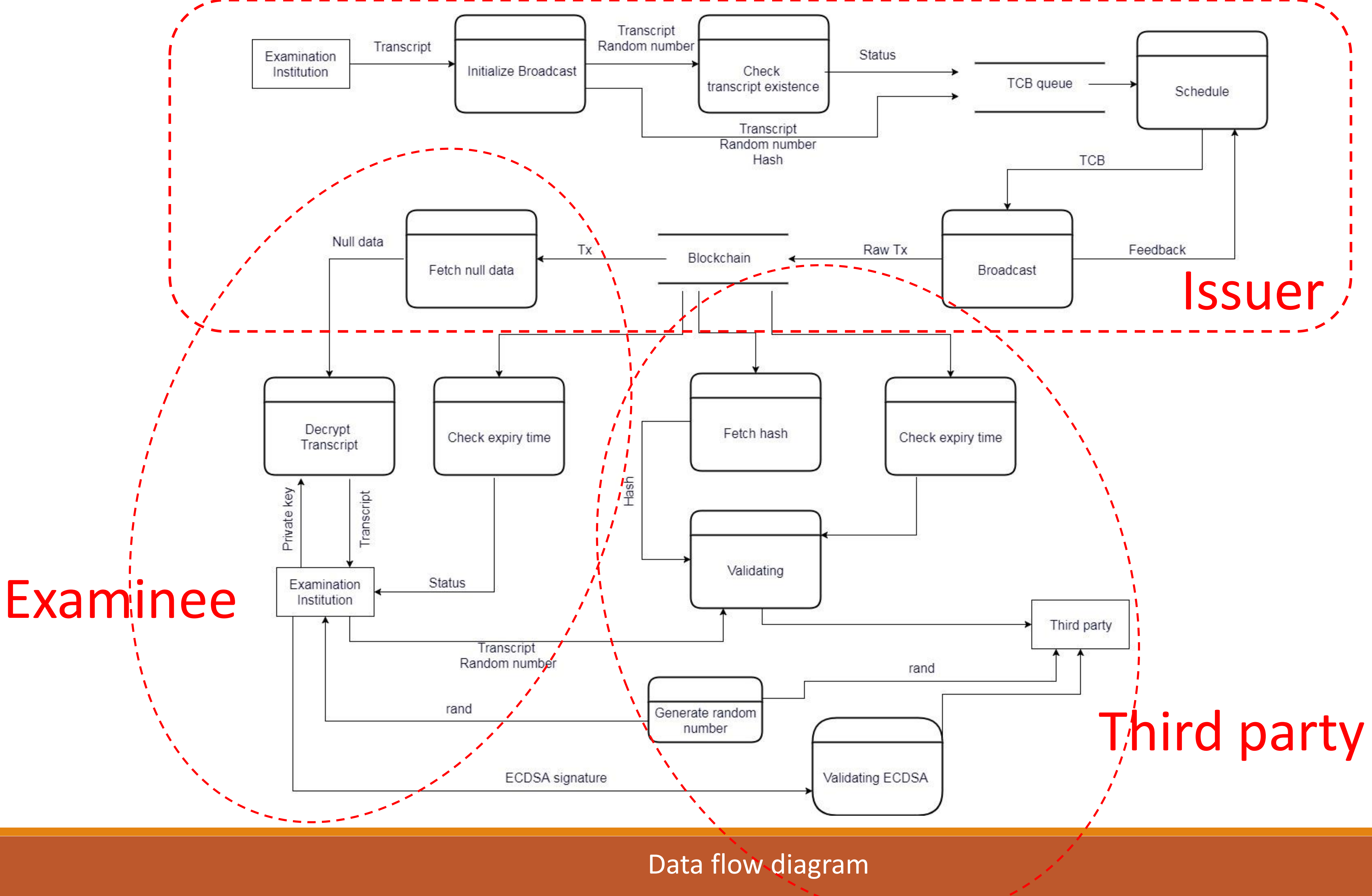
References



Data flow diagram

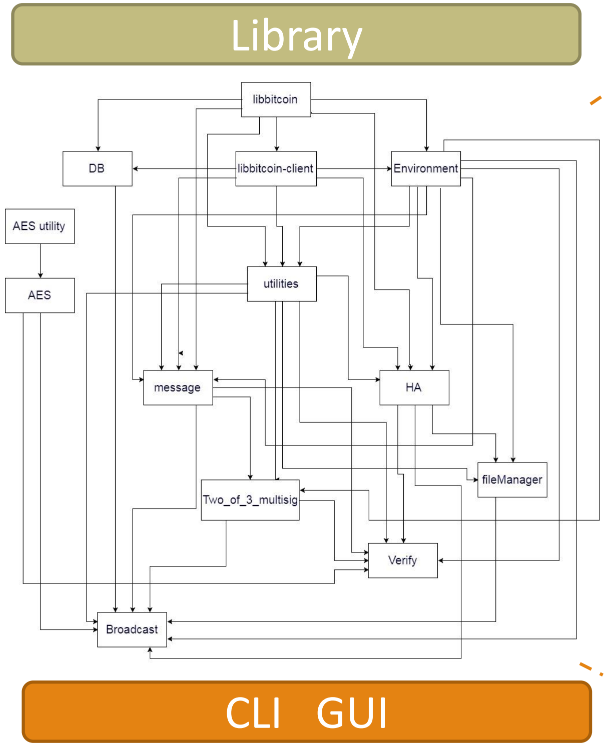


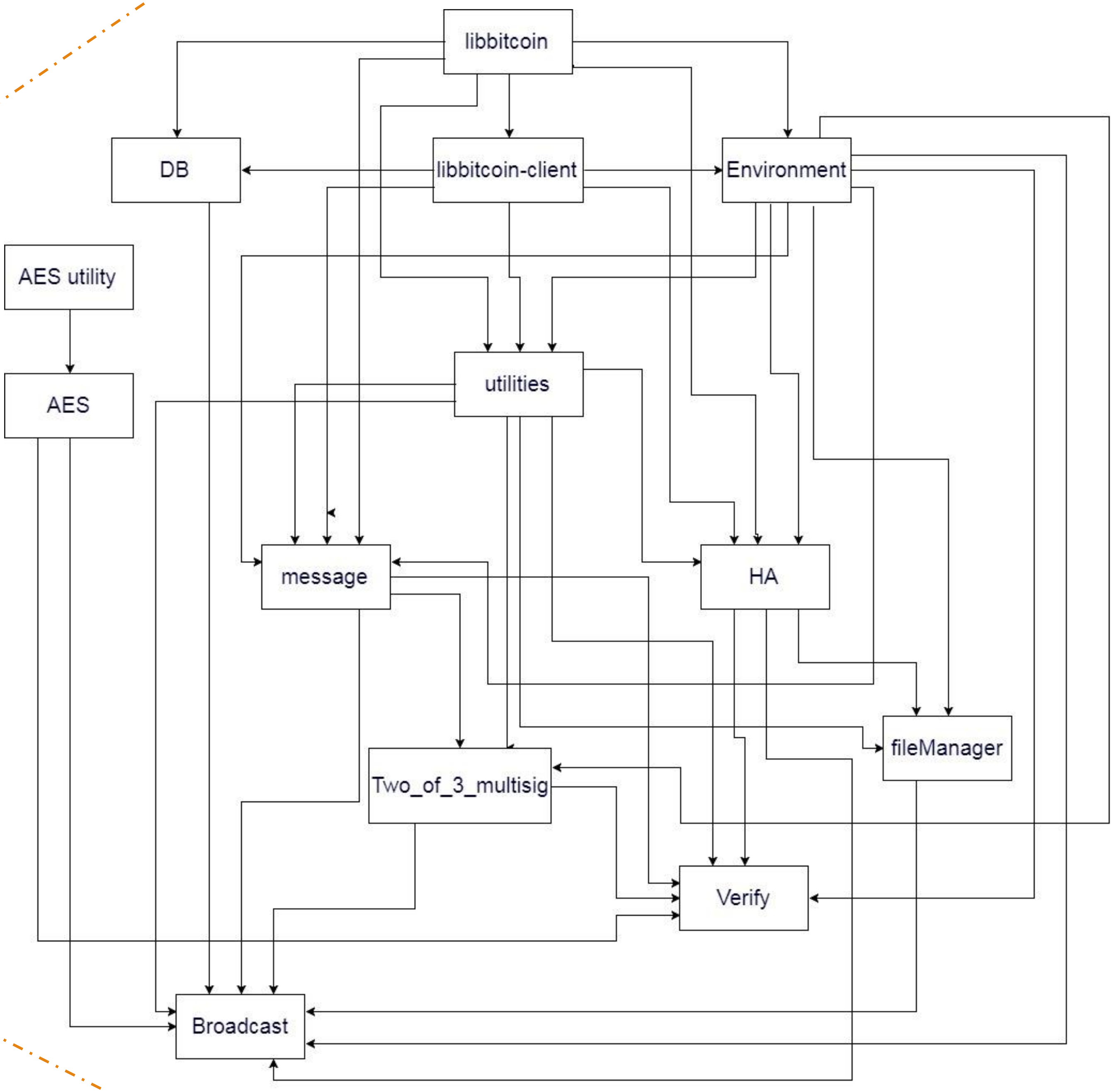
Data flow diagram



Data flow diagram

# Implementation:

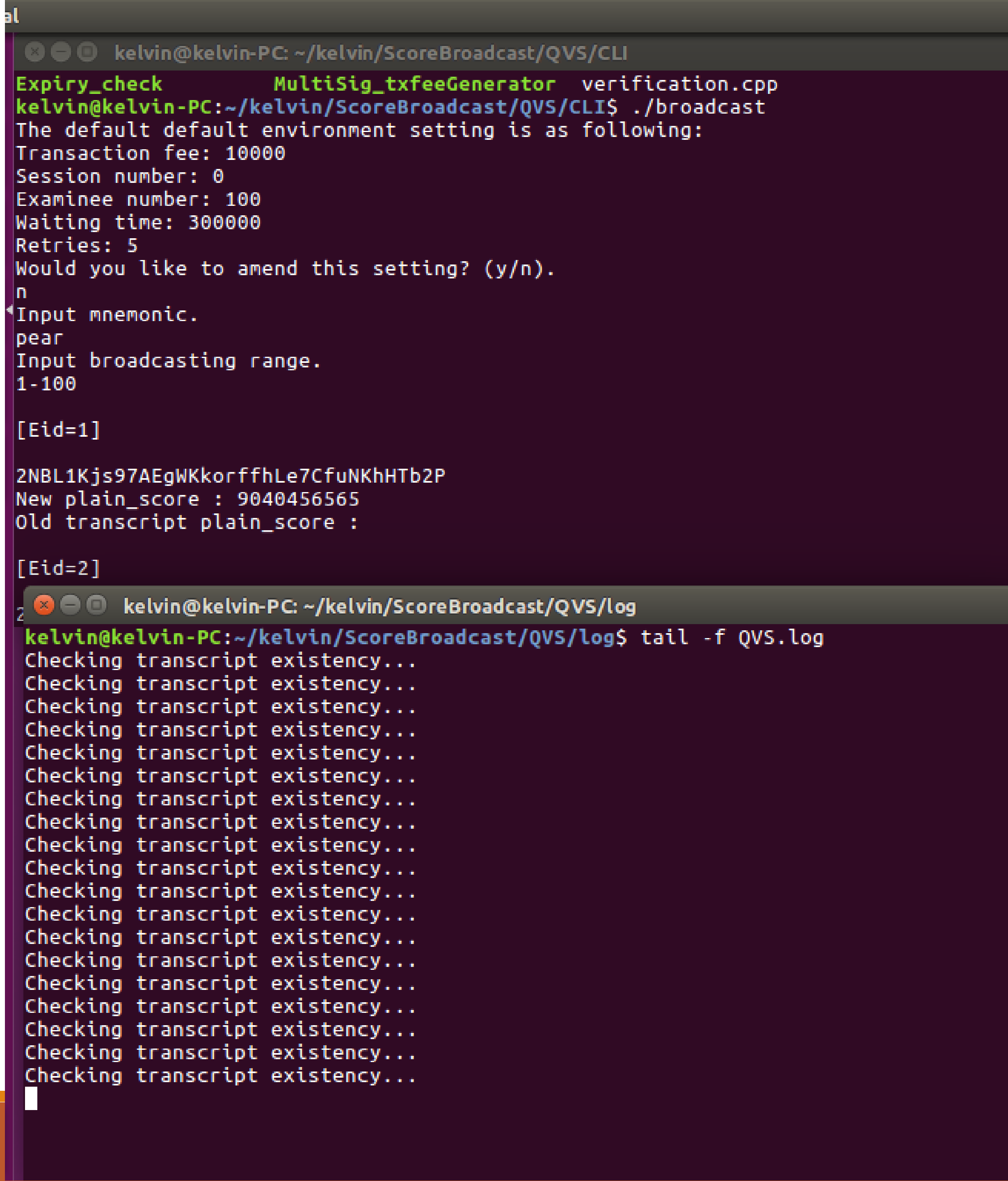




Dependency relation

# Broadcasting time complexity:

| Number of examinees | Time (minutes) |
|---|---|
| 1 | 13.28 |
| 10 | 8.53 |
| 1000 | 33.71 |
| 10000 | 95.69 |
| | |



```
Expiry_check          MultiSig_txfeeGenerator  verification.cpp
kelvin@kelvin-PC:~/kelvin/ScoreBroadcast/QVS/CLI$ ./broadcast
The default default environment setting is as following:
Transaction fee: 10000
Session number: 0
Examinee number: 100
Waiting time: 300000
Retries: 5
Would you like to amend this setting? (y/n).
n
Input mnemonic.
pear
Input broadcasting range.
1-100

[Eid=1]

2NBL1Kjs97AEgWKkorffhLe7CfuNKhHTb2P
New plain_score : 9040456565
Old transcript plain_score :

[Eid=2]
2
```

```
kelvin@kelvin-PC:~/kelvin/ScoreBroadcast/QVS/log$ tail -f QVS.log
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
Checking transcript existency...
```

| no. of bits ($b$) | hash space size ($2^b$) | Number of hashed elements such that probability of at least one hash collision ≥ $p$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | $p = 10^{-18}$ | $p = 10^{-15}$ | $p = 10^{-12}$ | $p = 10^{-9}$ | $p = 10^{-6}$ | $p = 0.001$ | $p = 0.01$ | $p = 0.25$ | $p = 0.50$ | $p = 0.75$ |
| 32 | $4.3\times10^{9}$ | 2 | 2 | 2 | 2.9 | 93 | $2.9\times10^{3}$ | $9.3\times10^{3}$ | $5.0\times10^{4}$ | $7.7\times10^{4}$ | $1.1\times10^{5}$ |
| (40) | $1.1\times10^{12}$ | 2 | 2 | 2 | 47 | $1.5\times10^{3}$ | $4.7\times10^{4}$ | $1.5\times10^{5}$ | $8.0\times10^{5}$ | $1.2\times10^{6}$ | $1.7\times10^{6}$ |
| (48) | $2.8\times10^{14}$ | 2 | 2 | 24 | $7.5\times10^{2}$ | $2.4\times10^{4}$ | $7.5\times10^{5}$ | $2.4\times10^{6}$ | $1.3\times10^{7}$ | $2.0\times10^{7}$ | $2.8\times10^{7}$ |
| 64 | $1.8\times10^{19}$ | 6.1 | $1.9\times10^{2}$ | $6.1\times10^{3}$ | $1.9\times10^{5}$ | $6.1\times10^{6}$ | $1.9\times10^{8}$ | $6.1\times10^{8}$ | $3.3\times10^{9}$ | $5.1\times10^{9}$ | $7.2\times10^{9}$ |
| (96) | $7.9\times10^{28}$ | $4.0\times10^{5}$ | $1.3\times10^{7}$ | $4.0\times10^{8}$ | $1.3\times10^{10}$ | $4.0\times10^{11}$ | $1.3\times10^{13}$ | $4.0\times10^{13}$ | $2.1\times10^{14}$ | $3.3\times10^{14}$ | $4.7\times10^{14}$ |
| 128 | $3.4\times10^{38}$ | $2.6\times10^{10}$ | $8.2\times10^{11}$ | $2.6\times10^{13}$ | $8.2\times10^{14}$ | $2.6\times10^{16}$ | $8.3\times10^{17}$ | $2.6\times10^{18}$ | $1.4\times10^{19}$ | $2.2\times10^{19}$ | $3.1\times10^{19}$ |
| (192) | $6.3\times10^{57}$ | $1.1\times10^{20}$ | $3.5\times10^{21}$ | $1.1\times10^{23}$ | $3.5\times10^{24}$ | $1.1\times10^{26}$ | $3.5\times10^{27}$ | $1.1\times10^{28}$ | $6.0\times10^{28}$ | $9.3\times10^{28}$ | $1.3\times10^{29}$ |
| 256 | $1.2\times10^{77}$ | $4.8\times10^{29}$ | $1.5\times10^{31}$ | $4.8\times10^{32}$ | $1.5\times10^{34}$ | $4.8\times10^{35}$ | $1.5\times10^{37}$ | $4.8\times10^{37}$ | $2.6\times10^{38}$ | $4.0\times10^{38}$ | $5.7\times10^{38}$ |
| (384) | $3.9\times10^{115}$ | $8.9\times10^{48}$ | $2.8\times10^{50}$ | $8.9\times10^{51}$ | $2.8\times10^{53}$ | $8.9\times10^{54}$ | $2.8\times10^{56}$ | $8.9\times10^{56}$ | $4.8\times10^{57}$ | $7.4\times10^{57}$ | **$1.0\times10^{58}$** |
| 512 | $1.3\times10^{154}$ | $1.6\times10^{68}$ | $5.2\times10^{69}$ | $1.6\times10^{71}$ | $5.2\times10^{72}$ | $1.6\times10^{74}$ | $5.2\times10^{75}$ | $1.6\times10^{76}$ | $8.8\times10^{76}$ | $1.4\times10^{77}$ | $1.9\times10^{77}$ |

## Collision probability table

# Results and discussion:

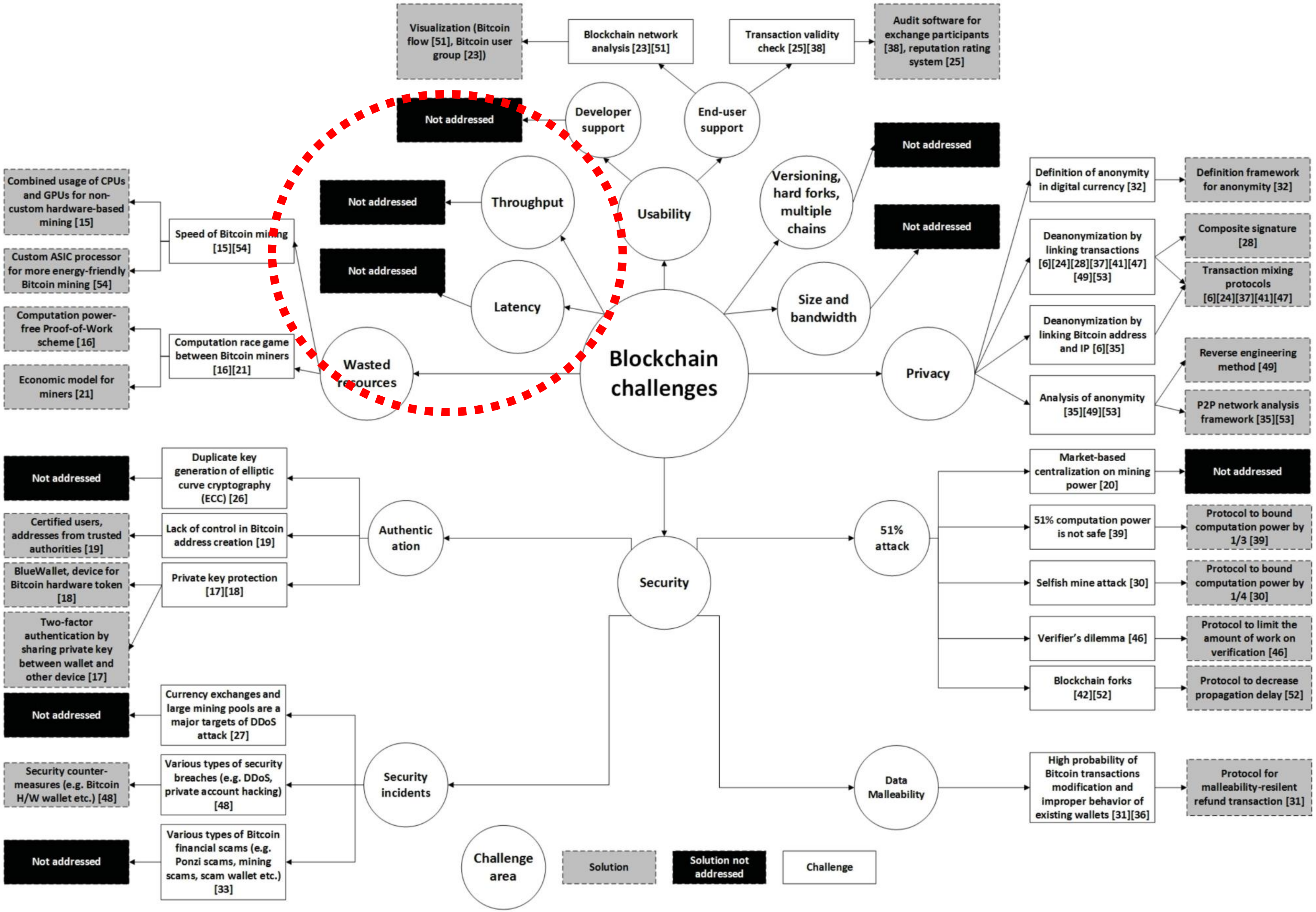
## Strengths of GIVS:

- Open to everyone(Adaptability)
- Tamper-proof (Immutable)
- Reliable, stable
- Protect the privacy of users
- Simple to check the validity of transcripts
- Convenient and easy to use
- Lower cost than traditional systems
- Information will be kept forever

# Issues and future work:

## Defects inherited from Bitcoin:

- Throughput → Only 1 Megabytes for each block
- Latency → New blocks are generated every 10 minutes



Blockchain challenges

Usability

Developer support

Not addressed

End-user support

Blockchain network analysis [23][51]

Visualization (Bitcoin flow [51], Bitcoin user group [23])

Transaction validity check [25][38]

Audit software for exchange participants [38], reputation rating system [25]

Throughput

Not addressed

Latency

Not addressed

Wasted resources

Speed of Bitcoin mining [15][54]

Combined usage of CPUs and GPUs for non-custom hardware-based mining [15]

Custom ASIC processor for more energy-friendly Bitcoin mining [54]

Computation race game between Bitcoin miners [16][21]

Computation power-free Proof-of-Work scheme [16]

Economic model for miners [21]

Versioning, hard forks, multiple chains

Not addressed

Size and bandwidth

Not addressed

Privacy

Definition of anonymity in digital currency [32]

Definition framework for anonymity [32]

Deanonymization by linking transactions [6][24][28][37][41][47][49][53]

Composite signature [28]

Transaction mixing protocols [6][24][37][41][47]

Deanonymization by linking Bitcoin address and IP [6][35]

Analysis of anonymity [35][49][53]

Reverse engineering method [49]

P2P network analysis framework [35][53]

Security

Authentication

Duplicate key generation of elliptic curve cryptography (ECC) [26]

Not addressed

Lack of control in Bitcoin address creation [19]

Certified users, addresses from trusted authorities [19]

Private key protection [17][18]

BlueWallet, device for Bitcoin hardware token [18]

Two-factor authentication by sharing private key between wallet and other device [17]

Security incidents

Currency exchanges and large mining pools are a major targets of DDoS attack [27]

Not addressed

Various types of security breaches (e.g. DDoS, private account hacking) [48]

Security counter-measures (e.g. Bitcoin H/W wallet etc.) [48]

Various types of Bitcoin financial scams (e.g. Ponzi scams, mining scams, scam wallet etc.) [33]

Not addressed

51% attack

Market-based centralization on mining power [20]

Not addressed

51% computation power is not safe [39]

Protocol to bound computation power by 1/3 [39]

Selfish mine attack [30]

Protocol to bound computation power by 1/4 [30]

Verifier's dilemma [46]

Protocol to limit the amount of work on verification [46]

Blockchain forks [42][52]

Protocol to decrease propagation delay [52]

Data Malleability

High probability of Bitcoin transactions modification and improper behavior of existing wallets [31][36]

Protocol for malleability-resilient refund transaction [31]

Challenge area

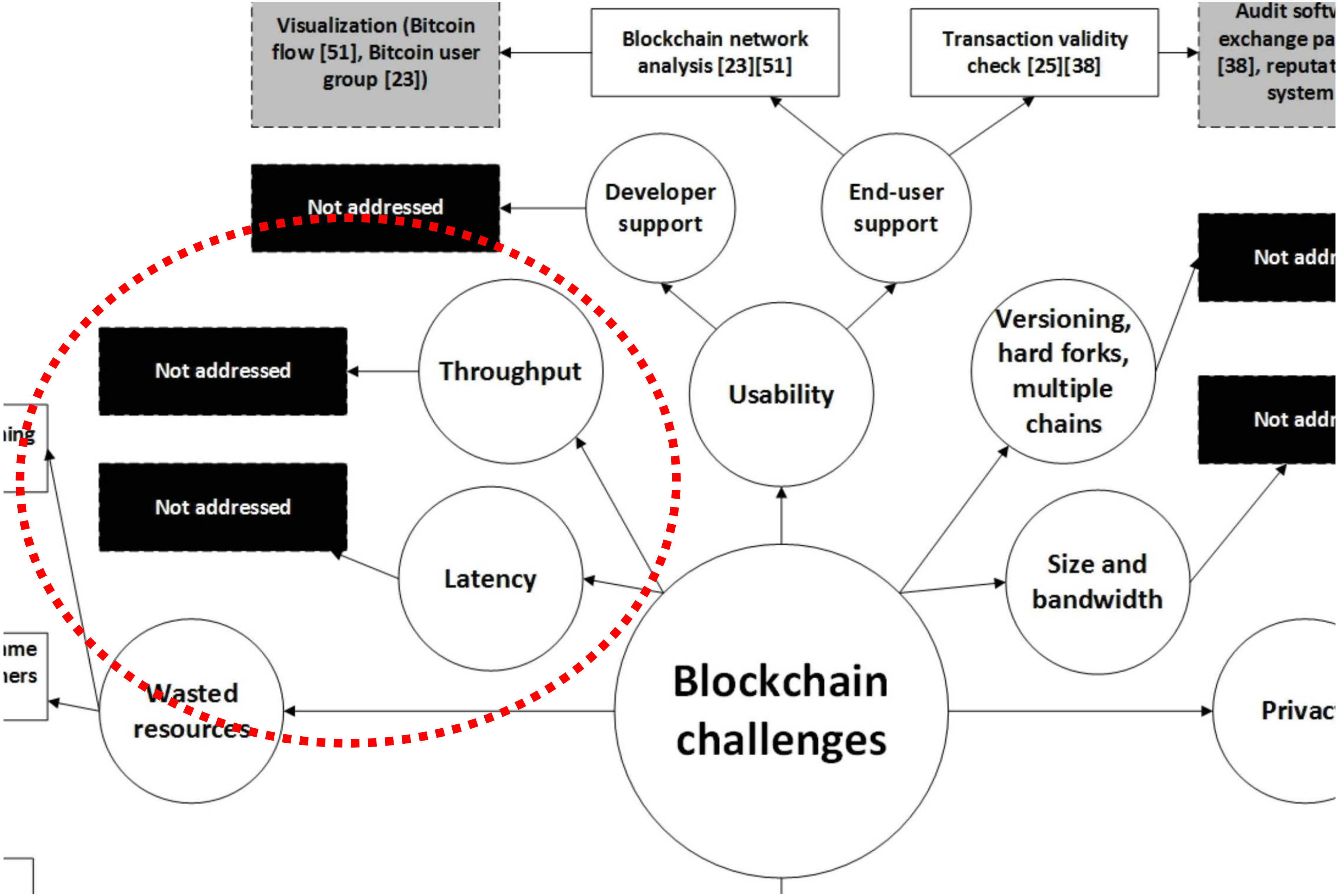
Solution

Solution not addressed

Challenge

October 3,2016

Citation: Yli-Huumo J, Ko D, Choi S, Park S, Smolander K (2016)Where Is Current Research on Blockchain Technology


Visualization (Bitcoin flow [51], Bitcoin user group [23])
Blockchain network analysis [23][51]
Transaction validity check [25][38]
Audit softw exchange pa [38], reputat system
Not addressed
Developer support
End-user support
Not addr
Not addressed
Throughput
Usability
Versioning, hard forks, multiple chains
ing
Not addr
Not addressed
Latency
Size and bandwidth
me ers
Wasted resources
Blockchain challenges
Privac

October 3,2016

Citation: Yli-Huumo J, Ko D, Choi S, Park S, Smolander K (2016)Where Is Current Research on Blockchain Technology

# References:

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”,http://www.bitcoin.org, (2008).

[2] Narayanan, Arvind, “Bitcoin and cryptocurrency technologies a comprehensive introduction.” (2016).

[3] Antonopoulos, Andreas M. M. “ Mastering Bitcoin.” O'Reilly Media, (2014).

[4] Daemen, Joan; Rijmen, Vincent (March 9, 2003). "AES Proposal: Rijndael" . National Institute of Standards and Technology. p. 1. Retrieved 21 February (2013).

[5] Stallings, William. “Cryptography and network security : principles and practice”, (2011).

# Thanks for your time!

# Q & A