Computational Cost Reduction of Transaction Signing in Blockchain

Kiattikul Sooksomsatarn[1], Ian Welch[2], Sakorn Mekruksavanich[3], Jenasama Srihirun[4]

[1,3] Faculty of Information and Communication Technology, University of Phayao, Thailand,

[2] School of Engineering and Computer Science, University of Wellington, New Zealand

[4] Khon Kaen Business School (KKBS), Faculty of Business Administration and Accountancy, Khon Kaen University, Khon Kaen, Thailand

[1] ajkiattikul@gmail.com, [2]ian.welch@gmail.com, [3] sakorn.me@up.ac.th, [4] jenasama.srihirun@gmail.com)

ABSTRACT

Nowadays, Blockchain is a disruptive technology, particularly in the financial context. Moreover, Blockchain is behind the success of cryptocurrencies, e.g., Bitcoin and Ethereum. Unlike traditional currencies, cryptocurrencies are entirely virtual. There is no physical money, but it can directly make payments in digital currency from one person to another without intermediaries. Moreover, Hashing's cryptographic algorithm makes Blockchain resist tampering from any transacting participants because the submitted block cannot be altered or re-engineered. However, another big problem is how users of cryptocurrencies stop somebody from adding or editing a transaction that spends someone else's money to them. To do this, Blockchain needs another cryptosystem called Public/Private Keys, a primitive asymmetric cryptosystem, e.g., the RSA encryption, to sign the transactions for proving the authenticity of the ownership without revealing the signed secret information. The generated public key is regarded as a ledger account number or digital wallet of the sender and the recipient. Simultaneously, the paired private keys are used to identify whether the digital wallets' owners are authentic. As growing network entities and propagated Blockchain transactions, computing millions of replicated tokens in the blocks to sign and verify the digital wallet's ownership is computationally expensive. However, a certain of chosen arithmetical transformations that can simplify mathematical cost can significantly reduce computational complexity. This research's main contribution is developing a protocol that can reduce the complexity and mathematical cost in generating the digital wallet and verifying its authenticity of ownership. Finally, performance analyses of the RSA algorithm for the protocol have been measured and visualized using Python.

Index Terms

Complexity, Blockchain, Bitcoin, Cryptocurrency, Digital currency, Public/Private Keys, Asymmetric Cryptosystem, the Rivest-Shamir-Adleman Encryption (RSA).

Article Received: 10 August 2020, Revised: 25 October 2020, Accepted: 18 November 2020

Introduction

Nowadays, Blockchain is a disruptive technology. It is a shared database storage technology or known as Distributed Ledger Technology (DLT). Replicated data logging guarantees that the previously recorded data is safe, seeing the same data from all network users and cannot be converted or edited. Such a mechanism provides the reliability of shared data using cryptography principles and the ability of distributed computing.

The start of Blockchain technology firstly took place in 2008 with Satoshi Nakamoto's presentation [1]. It is a conceptual expression of creating a secure platform that can create security in exchanging digital currency named Bitcoin. It is unnecessary for intermediaries, such as banks or other entities involved in the payment process. It receives widespread attention and acceptance from experts worldwide that it is technologically advanced and capable of using financial and banking aspects and other major sectors, including the government, science, and education sectors.

The Blockchain technology links all information together throughout the system. When a new transaction occurs, it must be announced to every machine in the system to recognize it. The transaction must also receive a consensus from the entire network before the transaction can be recorded to the Block. Therefore, Blockchain technology does not require an intermediary to do the job of organizing it. All the information is stored under the Blockchain structure and distributed to the local affiliate members. If someone tries to create a fake transaction, the data will conflict with the other members' device's data since the device connects to all other data, leading to tolerance to create such a transaction to the system. There will only be transactions that everyone in the affiliate accepts that can be recorded. The registered transactions that have been logged into the Blockchain system cannot be changed or modified. It is the main reason that makes Blockchain technology widely accepted as technology storage with high reliability.

The components of Blockchain consist of four essential elements: 1) Block 2) Chain 3) Consensus and 4) Validation. Blockchain data entry is calculated in Block type layout, and each Block links to the previous Block with Block's Hash value. They are always cascaded together into chains, making it difficult to counterfeit. They can check the data blocks' integrity throughout the Chain, which can be traced back through the starting Block called Genesis Block.

Block is a data packet divided into two parts. The first one is a Block Header part of letting us know what is inside the data box. The second one is a Block Data part to contain various information, such as information on the amount of money, transfer information, medical history information, and other miscellaneous information.

Chain is the principle of memorizing every transaction of everyone in the system, recording the data, copying the ledger, and distributing the ledger account to its people. The system will distribute a copy of the ledger account to everyone in the network to be aware of the conflict transactions. Since the launch of the Blockchain system, even if one node is corrupted, it can confirm or re-enter the other node's valid ledger, contributing to the system's availability.

Consensus is an establishment of an agreement and mutual commitment between members of the Blockchain affiliate. All of the members must agree to the rules and a mechanism to control the data's accuracy in all nodes through different algorithms to ensure the same precision and integrity, including information. There are several Consensus processes, such as Proof-of-Work, Proof-of-Stake, Practical Byzantine Fault Tolerance (PBFT), and Proof-of-Authority.

Validation is an entire-system proof of validity and accuracy of data to ensure that no information is missing. The part of the Consensus is Proof-of-Work. Proof-of-Work is creating a Consensus using solving complex mathematic problems that takes a long time from nodes belonging to the network, known as Miners. The goal of Proof-of-Work is to verify the reliability of the information being logged into the network. Miner will be rewarded for doing Proof-of-Work. Consequently, the Block will gradually be chained and is always available without any system administrator. Bitcoin is an excellent example of making a Consensus using a Proof-of-Work, which is a public Blockchain.

Although Blockchain technology is novel and cutting-edge, it is from nothing more than ancient knowledge called cryptography. There are two main cryptosystems used in Blockchain. The first one is Hashing, and the second one is Public/Private Keys.

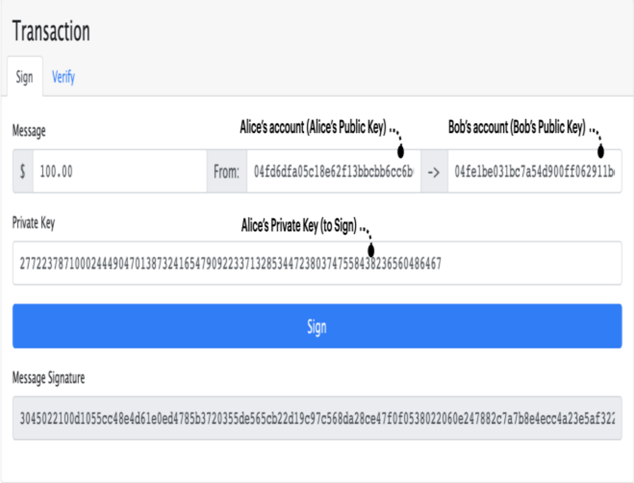
One of the most spread-used algorithms in Blockchain is Hashing. It is for creating a digital fingerprint of the data—the Hash changes if and only if the data changes. Hashing is also used for Mining. According to the specified conditions, the miner who can first find the suitable Hash with the corresponding "number only used once" (Nonce) will be rewarded. The designation that every node has to find the Nonce by solving a mathematical problem, known as Proof-of-Work, where the accepted Block must answer the Block's aforementioned mathematical problems. Therefore, a malicious attacker cannot create a fake transaction without modifying Nonce's mathematical answer, including Block and all other Chain's blocks. Moreover, they must be done simultaneously in every Block to be accepted by other nodes, called the Consensus. That is difficult and almost impossible with the ability of today's computers. Consequently, Blockchain is regarded as a form of recording highly secure data.

Another cryptographic primitive used in Blockchain is Public/Private Keys. It can be used not only for creating a digital wallet but signing transactions as well. A keypair generator creates digital keys. The public key is the address of the wallet, which is approximately 512 character-long hexadecimal characters. These numbers are used as a bank account number of the sender and the recipient. Simultaneously, the paired Private key is used to sign and verify the ownership of the wallet. The private key must be first generated using random numbers, but it needs to be kept secret as the name implies. Only the authentic wallet's owner knows the private key. After that, the public key is generated by using the chosen private key encrypted with asymmetric algorithm like RSA. Anyone in the network can know the public key because it is regarded as the wallet number. Finally, the cryptosystem can prove knowledge of a fake transaction without revealing the private key. In other words, the cryptographic mechanism can detect whether the transaction was tempered without using the private key. Consequently, the malicious attacker cannot change the signed transaction information because even if they know the sender's and the recipient's public key, they do not know the private key to re-sign it. The digital signature will change immediately when the malicious node changes the information in the transaction.

The wallet's ownership is established through digital keys, the wallet's address, and digital signature. The digital keys are not stored in the network but are instead created and stored by users in a simple file called a wallet. Most transactions require a valid digital signature in the Blockchain, which can only be generated with the owner's private key. Keys come in pairs consisting of a private (secret) key and a private key. The public key is similar to a ledger number, and the private key is similar to the secret PIN or signature on a cheque.

In practice, the popular cryptocurrency like Bitcoin is structured as a peer-to-peer network topology. The practical Bitcoin network consists of between 5,000 to 8,000 listening nodes running various Bitcoin reference clients [2]. All nodes include the routing function to participate in the network and might include other supporting functionality, e.g., the Blockchain database, mining, and wallet services. All nodes provide to validate and propagate transactions and blocks—accordingly, the more extensive network and longer propagated blocks, the more transactions to be validated.

A transaction consists of four pieces of information to be signed: an amount of payment, sender's account, recipient's account, and sender's private key. Then, when the information is signed, the signed transaction (message signature) is sent to the recipient with three public information pieces. Meanwhile, the private key is kept secret with the sender. The web-based transaction signature simulation is shown in Fig. 1.


Transaction
Sign
Verify
Alice's account (Alice's Public Key)
Bob's account (Bob's Public Key)
Alice's Private Key (to Sign)
Sign

Message

$ 100.00 From: 04fd6dfa05c18e62f13bbcbb6cc6b -> 04fe1be031bc7a54d900ff062911b

Private Key

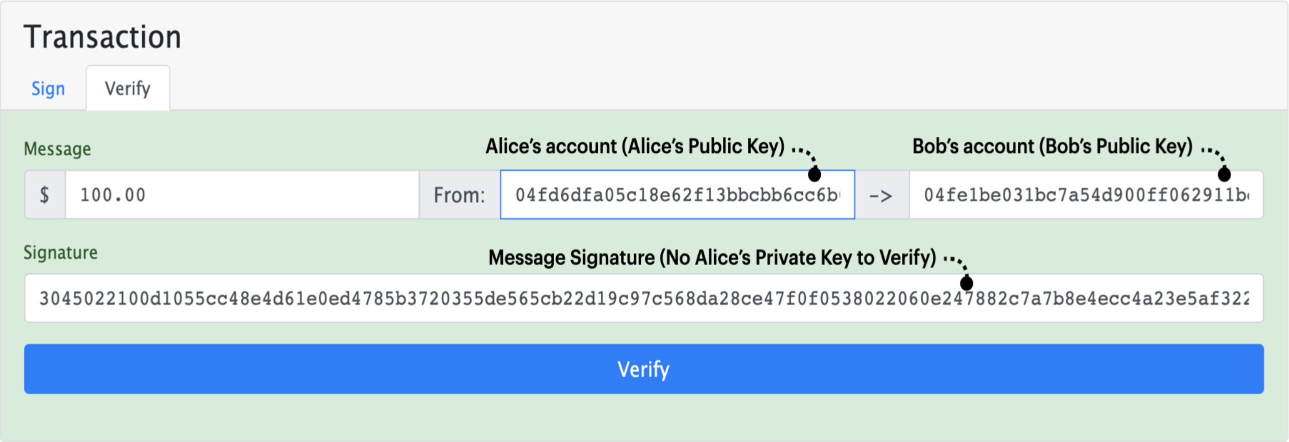
27722378710002444904701387324165479092233713285344723803747558438236560486467

Message Signature

3045022100d1055cc48e4d61e0ed4785b3720355de565cb22d19c97c568da28ce47f0f0538022060e247882c7a7b8e4ecc4a23e5af32

Fig.1 Alice's transaction to Bob's account

Fig. 2 illustrates when the recipient obtains the signed transaction. The recipient can verify whether the received transaction was altered using four pieces of information: the amount of payment, sender's account, recipient's account, and the signed transaction.



Transaction

Sign Verify

Message Alice's account (Alice's Public Key) Bob's account (Bob's Public Key)

$ 100.00 From: 04fd6dfa05c18e62f13bbcbb6cc6b -> 04fe1be031bc7a54d900ff062911b

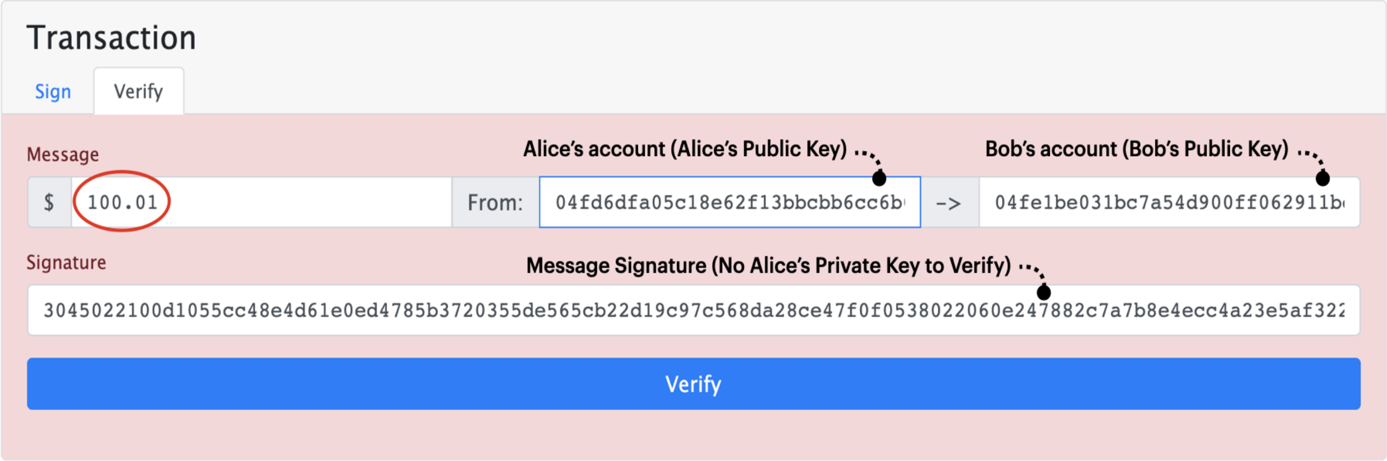
Signature Message Signature (No Alice's Private Key to Verify)

3045022100d1055cc48e4d61e0ed4785b3720355de565cb22d19c97c568da28ce47f0f0538022060e247882c7a7b8e4ecc4a23e5af322

Verify

Fig.2 Alice's transaction verification

If any of the information was modified, the system could immediately perceive and inform all nodes in the network that the transaction is fake. Then, the fake transactions are dropped. For example, if a malicious node changes the amount of payment, even a cent, the screen will turn red which means the verification failed, as shown in Fig. 3.



Transaction

Sign Verify

Message Alice's account (Alice's Public Key) Bob's account (Bob's Public Key)

$ 100.01 From: 04fd6dfa05c18e62f13bbcbb6cc6b -> 04fe1be031bc7a54d900ff062911b

Signature Message Signature (No Alice's Private Key to Verify)

3045022100d1055cc48e4d61e0ed4785b3720355de565cb22d19c97c568da28ce47f0f0538022060e247882c7a7b8e4ecc4a23e5af322

Verify

Fig.3 Amount of payment changed, Alice's transaction verification failed

Transactions are the most important part of the cryptocurrency system. Everything else in cryptocurrency is designed to ensure that transactions can be created, propagated on the network, validated, and finally added to the global ledger of transactions (the Blockchain).

Therefore, researchers have realized the importance of bandwidth overhead spent on signing and verifying transactions in Blockchain. Researchers believe that certain arithmetic transformations can reduce the computational cost of signing and verifying transactions. In the next section, researchers will define our contributed research objectives of our academic paper.

Research Objectives

The objectives of this research article are 1) to develop a protocol that can reduce the complexity and mathematical cost in signing and verifying Blockchain transactions, and 2) to analyze performance of our protocol measured and visualized using Python.

Research Methods

A. Research Process

The researchers developed a web-based transaction signature simulation using Python Flask to demonstrate signing and verifying Blockchain transactions. The processes of implementation consist of

1. Measuring runtime spent on specific numbers of transactions signing and verifying based on the 2048-bit public RSA key-size represented by 512 hexadecimal characters and the private RSA key-size varied between 5 and 512 hexadecimal characters, which is the typical range of key size for Bitcoin.
2. Simplifying the complexity of computational cost using certain arithmetical transformations and measuring them again.
3. Visualizing the summarizing of the findings by Python.
4. Analyzing our protocol's performance compared to the original schemes.

B. Rules of Our Protocol

In the first step of the research process, the researchers defined the security model for creating keypairs and signing transactions, which sk_A is Alice's private key, pk_A is Alice's public key, sk_B is Bob's private key, pk_B is Bob's public key, *m* is the amount of payment Alice sending to Bob, and tx_A is the transaction signed by Alice as shown in Fig. 4.

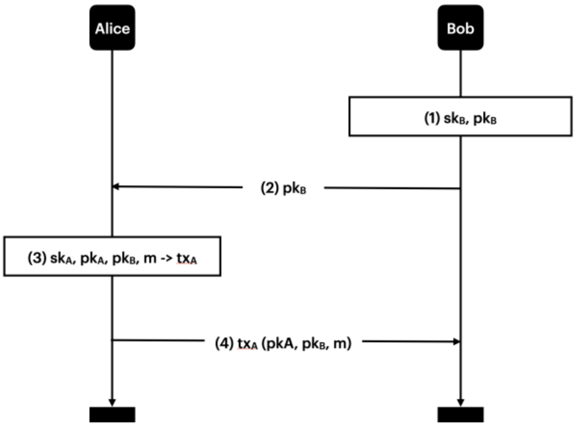

Alice
Bob
(1) sk_B, pk_B
(2) pk_B
(3) sk_A, pk_A, pk_B, m -> tx_A
(4) tx_A (pkA, pk_B, m)

Fig.4 Research process for keypair generations and signing transaction

The other security scheme is shown in Fig. 5, which is verifying transactions.

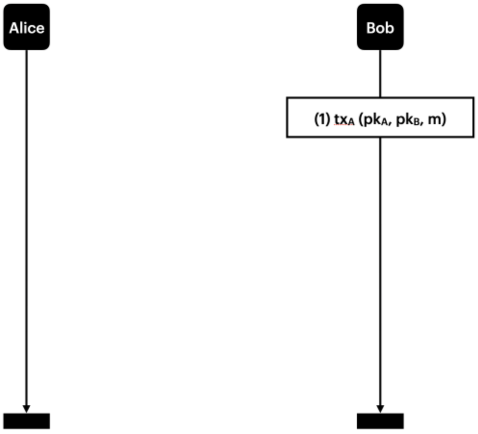


Fig.5 Research process for verifying transaction

In the second step of our research process, the RSA algorithm's modular exponents were simplified by certain arithmetical transformations. Although the formulas in each signing and verifying scheme are different, they concern with the Legendre symbol. Our protocol is implemented on the Legendre symbol and existing arithmetical transformations. We notice that the computational time of the Jacobi symbol increases exponentially when the key size increases. Therefore, our primary goal is reducing the "numerator" modulo the "denominator". The following are the rules of our algorithm. Note that the steps of each scheme are the same, but the private and public keys (the "numerator" and the "denominator")

1. Reformulating *the Legendre symbol* to *Euler's criterion*

$$\left(\frac{a}{p}\right) = a^{\left(\frac{p-1}{2}\right)} \pmod{p}$$

2. Reducing the "numerator" modulo the "denominator" using the *Extended Euclidean Algorithm*

3. Extracting any factors of 2 from the "numerator" using *the second supplement to the law of quadratic reciprocity*

$$\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}} = \begin{cases} 1, & \text{if } p \equiv 1 \text{ or } 7 \pmod{8} \\ -1, & \text{if } p \equiv 3 \text{ or } 5 \pmod{8}. \end{cases}$$

4. If the "numerator" is 1, it gives a result of 1. If the "numerator" and "denominator" are not coprime, it gives a result of 0

5. Otherwise, the "numerator" and "denominator" are now odd positive coprime integers so that we can calculate the time taken

The above algorithm leads to an efficient *O((log a)(log b))* [3] algorithm for calculating *the Jacobi symbol*, analogous to the *Euclidean algorithm* for finding *the greatest common divisor* of two number.

C. Experiment Machine and Tools

The experiments have been done on Dell Optiplex 9010 Quad Core i7-3570 3.40-3.80 GHz CPU, 16GB of RAM. The machine runs Python 3.8 on 64-bit Windows 10 Professional.

Results

This section explained the details of the implemented experiments compared with existing algorithms. The parameters considered to evaluate the protocol's performance are keys generation time, signing time, and verifying time. Each run was done ten thousand times, and the average and standard deviations are shown on the graphs.

Fig. 6 is the graph of time in seconds taken for using the RSA cryptosystem against the keys' size in numbers of digits. The value and x are fixed as five digits, and N varies from 128 to 512 digits.

- The green line showed the time taken without any arithmetic algorithm.
- The blue line showed the time taken using Jacobi symbol properties.
- The red line, almost on the X-axis, showed the time taken using our protocol.

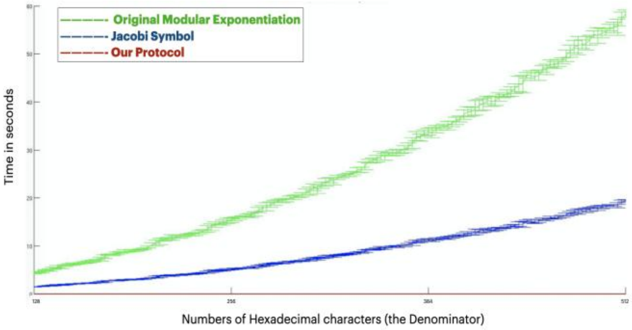


Fig.6 Time in seconds taken for signing and verifying 10,000
transactions with 5 Hexadecimal-digit private key and 128 to 512 Hexadecimal-digit public key compared with three schemes

To close up the graph plotted in Fig. 6. The runtime by our protocol varies from approximately 50 - 90 milliseconds, which is very small as shown in Fig. 7.

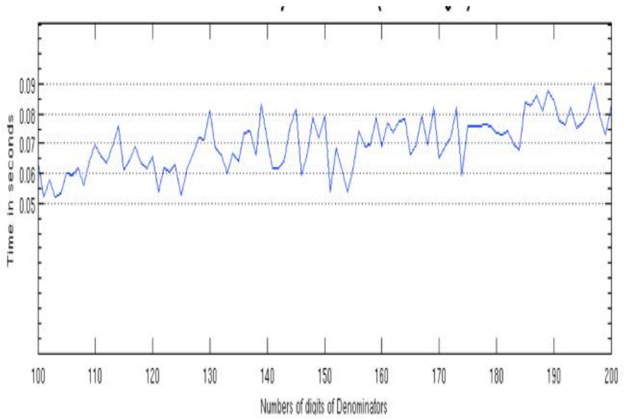


Fig.7 Time in seconds taken for signing and verifying 10,000
transactions with 5-digit private key and 100 to 200 digit public key of our protocol (closed-up)

In maximum, the numerator should be up to 512 digits. Therefore, we changed the number of digits of the numerator to 512 digits. The graph of the time taken of each algorithm is plotted in Fig. 8. The graphs of ordinary modular exponentiation significantly increased while the graph of both the Jacobi symbol algorithm and our protocol slightly increased in the time taken. However, our protocol's graph trended to be lower than the Jacobi symbol's when the numbers of digits of N increased.

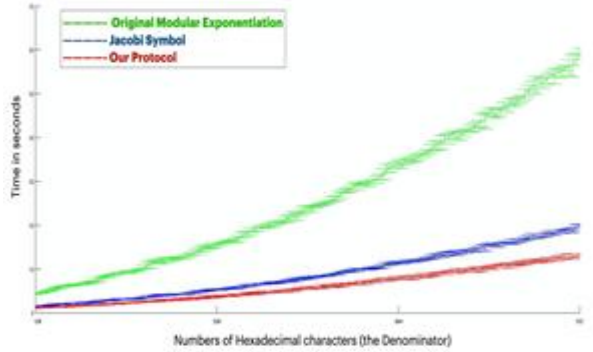


Fig.8 Time in seconds taken for signing and verifying 10,000 transactions with 512 Hexadecimal-digit private key and 128 to 512 Hexadecimal-digit public key compared with three schemes

Discussions

This section discussed our demonstrated protocol's performance analysis. The experimental results of the transaction signature procedures indicated that our protocol consumed the least amount of time compared to both the original modular exponential and the Jacobi symbol transformation, not only the small but also the large numbers of private keys. Moreover, it tended to be more significantly different from our scheme's slope and others' when the public key's numbers of digits increased. More importantly, our algorithm consumed at least 6 times less than the original one’s at the 512-digit key-size of the public key, or approximately 83.33% reduced. Additionally, it tended to be many more times when the public key's size was growing. As a result, our protocol can significantly reduce the computational cost for Blockchain's transaction signing and verifying.

Recommendations

The researchers have recommendations for Further Research as followings:
1. The experiment tested in this study were taken at a maximum 512-digit RSA key. The next research should be done longer key size to see how much our protocol has better performance than other schemes.
2. The experiment should be done with asymmetric-key encryption with another RSA-like algorithm, e.g., Goldwasser-Micali cryptosystem.
3. The computational cost should be reduced by another arithmetical transformation, e.g., Chinese Remainder’s Theorem or Fermat’s Little Theorem as well.

Conclusion

In conclusion, transaction signing and verification are regarded as the most critical part of Blockchain. The signed transaction guarantees the protection of a data modification to be safe and secure. The transactions tend to be more important as the propagation of blocks in the Blockchain contributed to the much more signing and verifying time consumption. The findings from the analyses of both small and larger key size of the private key showed that the runtime consumed by our algorithm is very compact. It is supposed to decrease much time to sign and verify transactions in practical Blockchain. Our contribution is computational cost reduction of transaction signing in Blockchain

References

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2009, from https://bitcoin.org/bitcoin.pdf.

[2] A. M. Antonopoulos, “Mastering Bitcoin: Programming the open Blockchain”. O’Reily Media Inc. USA, 2017.

[3] H. Cohen, “A Course in Computational Algebraic Number Theory”. Springer, 1993.