



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 1, January 2018

Blockchain based Security Mechanism for Internet of Things (IoT)

Zainiya Manjiyani ¹

Graduate Student, Department of Computer Science, Sacramento State University, Sacramento, California, USA¹

ABSTRACT: Today, as world is going towards automation and smart objects, IoT is most popular area for research right now. There are so many security issues related with IoT, for example, confidential data collection, insecure interfaces, unencrypted communications, etc. Traditional cryptography schemes are not good for the IoT as they put unnecessary overhead on low capacity IoT devices. Potential solutions to this problem are using lightweight cryptography or using existing cryptography blocks and modifying it to support IoT devices. In lightweight cryptography, we cannot lower the key size and cost of modern design and standard of modern design are issues. In existing cryptography block AES 256 can be used but it is still expensive for IoT device with low computing capability. The proposed solution is using blockchain to implement security mechanism for IoT devices. Each aggregator in network contains chain of transaction blocks replicated over peer to peer distributed network. Each block stores values from the sensors after applying hashing on it. The mechanism is custom blockchain, without features those are unnecessary for IoT.

KEYWORDS: Blockchain, Internet of Things, Security, Trust based algorithm

I. INTRODUCTION

Nowadays Internet of Things is a rapidly growing technology. With the advent of broadband internet, the number of devices connecting with internet is growing larger and larger. Devices are being connected to each other and functions as per the requirement or order they get from each other with minimal human interaction. In simple words, “connecting all devices to internet and operating them from any remote location via internet is Internet of Things” [4].

Internet of things has a threat of security because of “private data collection”, “insecure interfaces”, and “unencrypted communications”[4]. These issues arise when either system is designed badly or cryptography is implemented in wrong manner. We can claim system design as a bad design, if the platform is not able to handle underlying encryption technique or transactions done in the system are unencrypted or insecure. Cryptography can go wrong if essential part or function is removed to make it lighter to support processing capability of the hardware platform.

Traditional cryptography algorithms like RSA, or DES are so heavy for low capacity IoT devices. The properties we need in cryptographic algorithm are dynamic but verifiable insertion and deletion of new node in the network, authentication, security against bad behavior of node, less overhead, optional encryption, capable of handling data from different sources. The major requirement here is to having cryptography algorithm that supports security in peer to peer distributed network of microcontrollers. Microcontroller devices supports hashing. The following figure shows the runtime overhead put by RSA algorithm on microcontrollers.

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 6, Issue 1, January 2018

Arduino UNO	16Mhz AVR	==> 12596 ms*
Arduino Leonardo	16Mhz AVR	==> 12682 ms*
Arduino Mega	16Mhz AVR	==> 12596 ms*
Arduino Due	84Mhz ARM	==> 1032 ms*
Arduino Yún	16Mhz AVR + 400Mhz MIPS	==> 707 ms*
Intel Galileo	400Mhz x86	==> 192 ms*

Fig. 1. RSA RUNTIME OVERHEAD

Blockchain is a “distributed protocol” where “transaction history is verifiable”[4]. Blockchain based algorithm uses technique called proof of work to establish the security of the transaction history. It is mostly used in financial transactions handling in bitcoin system. Bitcoin based protocol uses public key infrastructure to sign each transaction and keeps ledger of all the transactions. It also allows anonymous devices to join or leave network anonymously.

For the IoT systems size of the ledger and anonymous joining and leaving of the network are undesirable. Network must be aware of newly added node and new node must be authenticated by verifying the history of its transactions. Typical blockchain blocks are composed of sequence of transactions that are signed and verified, hash of previous block, hash of current block, nonce, and timestamp.

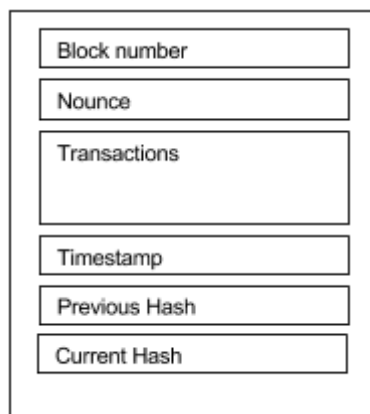


Fig. 2. Bitcoin Block

The Fig 2 shows the typical bitcoin block which contains block number, nonce, transaction ledger, timestamp, hash of previous block in chain and hash of current block in chain. Proof of work used here uses consensus algorithm approach. To understand consensus let us start with understanding hash and how it is used to create blockchain, then we can go in depth to understand verification method that verifies whether the node is bad node or not.

Hashing is a mathematical function that accepts data and applies some transformation and provides fixed length output as a hash value. The length of the hash value is not dependent on length of the data. But the value of the hash is highly dependent on the data, and with change in even single bit of data we can have huge variation in the hash value. In the consensus approach hash is created for the ledger of all the transactions available in the block in conjunction with nonce value and hash value from previous block. According to the system protocol, some signature value is decided such as first 32 bits of hash value must be 0 for the block to become valid block. So once the transactions are verified if any peer tries to change transaction record the hash value changes and it violates the signature of the system.

In the chain of blocks each block contains hash value from previous block as shown in Fig 3. If one of the block violates the transactions, it affects all the blocks following that bad block in the chain. Violated block can be

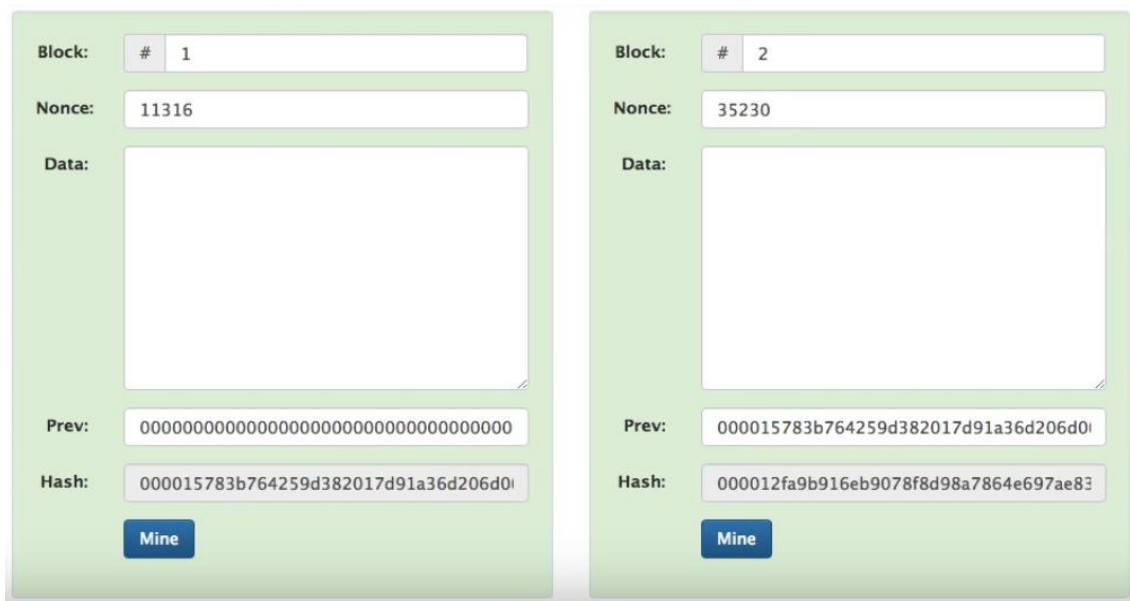
International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 1, January 2018

reconnected to chain by adjusting nonce value. But all the block following that node in the chain remains violated until the nonce value for all of them is adjusted.



Block #	Nonce	Prev Hash	Hash
1	11316	00000000000000000000000000000000	000015783b764259d382017d91a36d206d0
2	35230	000015783b764259d382017d91a36d206d0	000012fa9b916eb9078f8d98a7864e697ae83

Fig. 3. Chain of Block

As the violated block can rejoin the chain by adjusting nonce value how we can decide that which is bad node in peer network. To understand that first we have to study blockchain structure on all the peers in distributed network. In consensus blockchain systems blockchain is replicated on all the nodes in peer to peer network. When some violation and recovery happen on one peer it will change that values of blocks only on that node, all the other nodes will have the old values of hash and signatures. After detecting changes in signature on that single node, all the other node performs voting to decide whether those changes were ethical or not. If the result of voting proves changes as unethical the node is considered as bad node and disconnected from the network.

This consensus algorithm is suitable for financial transaction handling application where change in any historic transaction can lead to chaos and inconsistent database state. Once the transaction is done, no one is authorized to change or remove that. For the IoT there is not such a thing like history of transaction. In these systems, the validity of the transactions must be checked at the time when transaction takes place. There is no need of keeping the ledger of transaction of sensor values or other information that is exchanged over the network. Also, the hash function puts overhead of getting the signature value. If you want first 32 bit as 0 you have to check 232 combination of data which is 4,29,49,67,296 number of combinations and generating these many combinations on microcontroller-based system is huge overhead.

The proposed work suggests the use of distributed trust algorithm to identify malicious node and ensure the security of IoT systems. This work creates the code wrapper that uses distributed trust underneath the layer of trust checking and encrypted communication mechanism.

II. RELATED WORK

Since the general consensus algorithm is not applicable for the IOT devices, after thorough research, we decided to propose our new algorithm based on distributed trust network. Before going into the details of the algorithm, let us first get the background in distributed trust networks. Distributed trust networks are the networks, which are based on the principle of decentralization. There is no centralized nodes or decision makers in the network. It is a peer-to-peer

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 1, January 2018

communication-based network. In distributed trust network, any new node that comes in the network, is either considered trusted or distrusted. The variation to be used is totally dependent at the time of network setup. The trust networks are based on the variables of trust, mechanism of deciding the trust value and so on. The value of trust variable is obtained through mathematical computations. The mathematical computations are not global. They are local to a network and are decided at the time of setup of the network. Depending on the values of trust, it is decided whether a node is a bad node or valid node. If it is a bad node, the data of that node is no longer accepted and is discarded from the network. If the node is a good node, its trust value increases and over the time it gains the trust of all the nodes. The distributed trust networks are not IOT specific, they are deployed for large systems having very good computational resources. But the beauty of this model is that it gives the freedom of designing algorithms for the trust and so it is applicable to IOT devices as well where the computational power of the devices are not as much. For the systems, which have large computation powers we can have complex cryptography algorithms and thus security, bad node detection and ease of data transfer can be obtained. But the biggest challenge in IOT devices is to design an algorithm, which does not require large computation power but should be equally secure as well. Taking into consideration these perspectives, we propose the following algorithm.

III. PROPOSED ALGORITHM

The System architecture for the proposed algorithm is as follows:

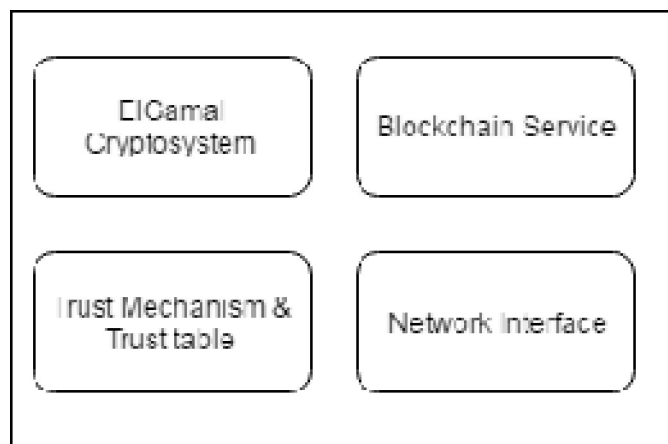


Fig 4. System Architecture

As shown in the Fig 4, our base will be a blockchain system running on a node, which essentially will be an IOT device. Our algorithm is focused on creating a network between such IOT devices having blockchain implementation to communicate amongst them. As described in the background study, traditional blockchain networking algorithms are not applicable for the IOT devices due to the less computation power. So, the proposed algorithm is the one, which is more feasible to be implemented on IOT devices.

The algorithm we propose here uses the ElGamal cryptosystem algorithm to generate public, private key and also for encryption and decryption. The reason of using ElGamal algorithm in place of RSA or SHA algorithm is the ease of generating keys using very less computational power. Also, the overhead is very less when we use ElGamal algorithm. The comparison and advantages of using ElGamal algorithm can be seen from the Fig 4.



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 1, January 2018

RSA	ElGamal
It is more efficient for encryption.	It is more efficient for decryption.
It is less efficient for decryption.	It is more efficient for decryption.
For a particular security level, lengthy keys are required in RSA.	For the same level of security, very short keys are required.
It is widely accepted and used.	It is new and not very popular in market.

Fig 5. RSA vs ElGamal

In our proposed algorithm every node will maintain a global key value pair table where the entries will be in the form of tuples. The tuples will have fields of an (unique id, public key, trust_value, trust_value_decreased_by). The table will look as shown in Fig 6.

ID	Public Key	Trust Value	Trust decreased by
AAA	aaa	50	CCC
BBE	bbb	52	N/A
CCC	ccc	48	BBB

Fig 6. Trust Table

Whenever a node joins the network, it will generate a public key using ElGamal algorithm. It will also have a hashing function which will take the input values of the node which are always static like manufacture id, warranty period, buy date and so on. It will generate an id using the hash function. The id and the public key will be given to the communicating neighbor node, which is already in the network. The node, which is in the network, will make an entry in the global table with the id, public key, trust value. The last value_decreased_by will only come in the picture when some node blames another node to be a bad node. The trust value initially will be 50% or half. The trust value will be increased or decreased as per the transactions. The data packet to transfer will have two fields header and payload. The header field will have the unique id, which is registered, in the global field. The id will be generated every time through the hashing function and if the static values have not changed the id will always remain the same. The whole packet will be encrypted with the receiving node public key, so that node can decrypt using its private key. While the procedure on the receiver side is a little different. On the receiver side, the node will first decrypt the packet using its private key. After decryption, it will check the id in packet with the global table, if it matches in the table entry, it will accept the data. If the id field does not match it will decrease the trust value of the node and will put its id in the trust_value_decreased_by field and send the updated tuple entry to all the nodes in the network. Whenever, some nodes' trust value is decreased, all the nodes will call the trust_process. In the trust_process, the node whose trust_value is decreased will be the candidate node while the other nodes will be the checker nodes. The checker nodes will make a remote procedure call to the candidate node which will just have the id of the candidate node encrypted with the candidate node's public key as its argument. The candidate node will then decrypt the argument with its private key, XOR the value with its public key, XOR the resulting value with its id and encrypt it with the checker node's public key and send it back. The checker node then will decrypt it with its private key and check the value. If the value comes out to be zero that means the bad node is not a bad node and the node, which has decreased the trust

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 6, Issue 1, January 2018

value, is the bad node. So, the checker node will increase the trust value of the candidate node and decrease the trust value of the node that has reported. If the value on the checker node side is not zero, then the checker node will decrease the trust_value further. So, at the end of trust process, the result will be out of who was originally the bad node.

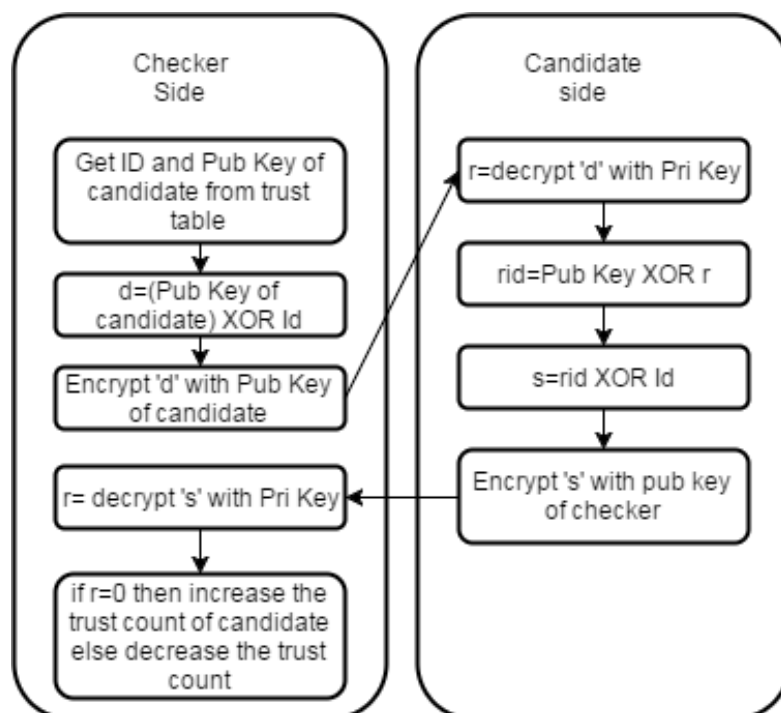


Fig 7. Trust Process

This implementation is very robust and scalable. With the trust procedure, we can actually find out the bad node. The trust process also takes care if the node, which is reporting the bad node, is itself a bad node, and then the innocent node should not be ousted. Here the only overhead, is maintaining the global table and updating the table in case of some bad node. Also, one more overhead will be the trust_value. The trust_value of any node is assigned half of the total number of nodes. So when, a new node enters, the trust value of the existing nodes is updated by one. Thus, the proposed algorithm is an ideal replacement for the consensus algorithm for blockchain implementation.

IV. RESULTS

The advantage of using this algorithm over probabilistic algorithm is the less overhead. In probabilistic algorithm, the overhead is very high as the probabilistic algorithm gives a new unique value to existing nodes at the time of the entry of the new node. The overhead of table updating increases with every entry of the node. Also, there is no proper mechanism for bad node detection. The bad node detection depends on the trust value and the trust value in probabilistic algorithm totally depends on the previous transactions. So, it also increases the overhead considering a node must do computations based on the previous transactions, every time a new transaction takes place. Also, the proposed algorithm is scalable. Any number of new node entries can be handled very effectively.



International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijirccce.com

Vol. 6, Issue 1, January 2018

V. CONCLUSION AND FUTURE WORK

Even though the algorithm is pretty much robust and scalable, there are still scopes, which can be improved for the future implementation. One of the assumptions that we made was that whenever a bad node is reported, we make remote procedure calls, but due to the limited recursive calls, only one report of bad node can be checked at once. Parallel trust procedures for two different nodes cannot be made. After one trust procedure ends, then and then only trust procedure for the other node can be started. This can be changed in future scope and a better way can be found to implement parallel trust procedure calls.

To find a scalable and robust algorithm with less overhead and implementable on devices with less computational power was a challenge in itself, but with the distributed trust network model as a background, we were able to propose a solution of our own. The proposed solution when implemented on IOT blockchain devices yields positive results. This algorithm can still be put under more research to improve its ability before its commercial implementation. Thus we can conclude that we have a proper blockchain based networking algorithm for IOT devices.

REFERENCES

1. Ali Dorri, S.S. Kanhere, R. Jurdak, "Towards an Optimized BlockChain for IoT", Proc. 2nd Int'l. Conf. Internet-of-Things Design and Implementation, pp. 173-78, 2017.
2. H. Yu, Z. Shen, C. Miao, C. Leung, and D. Niyato, "A survey of trust and reputation management systems in wireless communications" Proceedings of the IEEE, vol. 98, no. 10, pp. 1755–1772, 2010.
3. K. Christidies, M. Devetsiokiotis, "Blockchains and Smart Contracts for the Internet of Things", IEEEAccess on Special Section On the Plethera of Research in Internet of Things (IoT), 2016.
4. Konstantinos, K., Angelos, S., Irena, B., Jeff, V., & Tim, G. (n.d.). Leveraging Blockchain-based protocols in IoT systems.
5. L. Atzori, A. Iera, G. Morabito, "The internet of things: A survey", Comput. Netw., vol. 54, no. 15, pp. 2787-2805, Oct. 2010.
6. M. Younis, N. Krajewski, and O. Farrag, "Adaptive security provision for increased energy efficiency in wireless sensor networks," in 2009 IEEE 34th Conference on Local Computer Networks, Oct 2009, pp. 999–1005.
7. Morgan, J. (n.d.). A Simple Explanation Of 'The Internet Of Things'. Retrieved from <https://www.forbes.com/sites/jacobmorgan/2014/05/13/simple-explanation-internet-things-that-anyone-can-understand/#5380b3df1d09>
8. Neisse R., S. G. (2015, June 23). SecKit: A Model-based Security Toolkit for the Internet of Things. European Commission Joint Research Centre . Ispra, Ispra, Italy.
9. P. F. HaraldSundmaeker, P. Guillemin, S. Woelfflé, Vision and Challenges for Realising the Internet of Things" Pub. Office EU, 2010.
10. Srinivasan A., T. J. (2006). DRBTS: Distributed Reputation-based Beacon Trust System. Dependable, Autonomic and Secure Computing, 2nd IEEE International Symposium (p. 103). Indianapolis: IEEE.
11. Xu Xiaohui, Study on Security Problems and Key Technologies of The Internet of Things, 2013.
12. Z. Yan, P. Zhang, and A. V. Vasilakos, "A survey on trust management for internet of things," Journal of Network and Computer Applications, vol. 42, no. 120-134. 2014.