Implementing Bubbles

of Trust: A decentralized blockchain-based authentication system for IoT

Vibhor,Eyanshu Gupta

Department of Computer Science and Information Systems  
Birla Institute of Technology and Science, Pilani, Rajasthan, India  
{h20190517,h20190018}@pilani.bits-pilani.ac.in

***Abstract*—A number of challenges prevent the securing of IoT devices and ensuring end-to-end security in an IoT environment. Because the idea of networking appliances and other objects is relatively new, security has not always been considered top priority during a product's design phase. Additionally, because IoT is a nascent market, many product designers and manufacturers are more interested in getting their products to market quickly, rather than taking the necessary steps to build security in from the start.Security experts have long warned of the potential risk of large numbers of unsecured devices connected to the internet since the IoT concept first originated in the late 1990s.One of the most important emerging trends is the amalgamation of blockchain technology and the Internet of Things.The decentralization of an IoT network will provide it with the ability to solve a lot of its security challenges. Capabilities of the technology, including trustworthiness, decentralization, scalability, and autonomy, make it a potentially essential component of the overall IoT ecosystem**

***Keywords—IoT,Blockchain,Cryptography,Bubbles of Trust.***

# Introduction

IoT devices are used in multiple sectors and industries, including:Consumer applications – IoT consumer products include smartphones, smartwatches and smart homes, which control everything from air conditioning to door locks, all from a single device.Business applications – Businesses use a wide range of IoT devices, including smart security cameras, trackers for vehicles, ships and goods, as well as sensors that capture data about industrial machinery.Governmental applications – Governmental IoT applications include devices used to track wildlife, monitor traffic congestion and issue natural disaster alerts.IoT devices were not built with security in mind. In the majority of cases, there is no way to install security on the device itself. In addition, they sometimes ship with malware on them, which then infects the network they are connected to.Some network security doesn’t have the ability to detect IoT devices connected to it and/or the visibility to know what devices are communicating through the networkTherefore we need to set some basic parameters that need to be looked into such as.:How to prevent identity theft, data corruption, and use of fraudulent hardware?How to securely store operational parameters and keys on an IoT device that is often deployed in potentially harsh environments where keys can be compromised?How to securely store and communicate generated IoT application data to the larger IoT ecosystem?How to manage IoT device keys, configuration, and user credentials securely?etc.However most of this can be answered by implementing a public key infrastructure in combination with cryptography but however the resource constrained nature of IoT devices means the result will not be satisfactory.In order to run the complex nature of cryptography algorithms i.e encryption and decryption on a device that may hardly contains any powerful processing unit and to provide result in real time is really fruitless and time consuming on top using a public key infrastructure for authentication purposes which may in some case lead to time lag because of huge number device using the same PKI.

Consequently, it is necessary to propose new security solutions for the system-of- systems as a whole. The latter must: (1) allow an easy integra- tion of new devices as well as new services; (2) fully adapted to IoT requirements and needs; and (3) does not depend on the type of devices, nor on the use case architecture and design.

As many researchers have suggested that blockchains represent a very promising technology to meet security requirements in the IoT context. Benefiting from blockchains power and resiliency, in this work, we implement an efficient decentralized authentication mechanism called bubbles of trust as described in the research paper[1] .

# Related Work

[2]Christidis and Devetsikiotis (2016) provide a description of how blockchains and smart contracts can be integrated in IoT. They provide a list of the advantages and limits of blockchain use in IoT and conclude that using blockchains and smart contracts facilitates the sharing of IoT services and resources and allows the automation, in a cryptographically verifiable manner

[3]Malviya (2016) made a quick study on how blockchain features can secure the IoT. They also described some IoT platforms that rely on blockchains for numerous use case scenarios.

[4]Huh et al. (2017) propose an approach to integrate blockchains to IoT. Their approach relies on the idea of con- figuring each object by a dedicated smart contract that de- fines its actions.

[5]Bahga and Madisetti (2016) pro- pose a Blockchain Platform for Industrial Internet of Things (BPI- IoT) , that enhance the functionality of existing Cloud-Based Manufacturing (CBM) platforms, especially towards integrating legacy shop floor equipment into the cloud environment man- ufacturing. They also propose an architecture for IoT devices to support the proposed platform. However, to secure devices, they rely only on a key-pair, generated by the device itself, without any control. Thus, any user can exploit the system.

[6]Ruta et al. (2017) propose a novel Service-Oriented Architecture (SOA) based on a semantic blockchain for registration,discovery, selection and payment. Such operations are implemented as smart contracts, allowing distributed execution and trust.

[7]Dorri et al. (2016, 2017) propose a blockchain based archi- tecture for IoT. Their approach relies on three interconnected blockchains: a local blockchain (private) for each use case, a shared blockchain (private) and an overly blockchain (public). Even if the solution resolve the problem of identification, it has multiple shortcomings like the local blockchains are not distributed but centralized which is contrary to its principle because it can limit its power and availability.

[8]Ouaddah et al. (2016, 2017) propose FairAccess , a blockchain-based access control framework in IoT. More precisely, FairAccess works in the same way as Role-Based Access Control (RBAC) ( Ferraiolo et al., 1995 ), where the policies are stored in a private blockchain. Thus, the authenticity, the authentication and the updates of policies are always guar- anteed. However, it can handle only policy-based compatible systems and use cases, which cannot be applied to numerous IoT contexts.

# Background

This section gives a brief overview of the various technologies, platforms, and architectures that are used in the implementation of the proposed methodology.

## Blockchain

A blockchain is, in the simplest of terms, a time-stamped series of immutable records of data that is managed by a cluster of computers not owned by any single entity. Each of these blocks of data (i.e. block) is secured and bound to each other using cryptographic principles (i.e. chain).

The blockchain network has no central authority — it is the very definition of a democratized system. Since it is a shared and immutable ledger, the information in it is open for anyone and everyone to see. Hence, anything that is built on the blockchain is by its very nature transparent and everyone involved is accountable for their actions.

The blockchain is a simple yet ingenious way of passing information from A to B in a fully automated and safe manner. One party to a transaction initiates the process by creating a block. This block is verified by thousands, perhaps millions of computers distributed around the net. The verified block is added to a chain, which is stored across the net, creating not just a unique record, but a unique record with a unique history. Falsifying a single record would mean falsifying the entire chain in millions of instances. That is virtually impossible. Bitcoin uses this model for monetary transactions, but it can be deployed in many other ways.

Blockchain consists of three important concepts: blocks, nodes and miners.

Blocks:Every chain consists of multiple blocks and each block has three basic elements:

* The data in the block.
* A 32-bit whole number called a nonce. The nonce is randomly generated when a block is created, which then generates a block header hash.
* The hash is a 256-bit number wedded to the nonce. It must start with a huge number of zeroes (i.e., be extremely small).

When the first block of a chain is created, a nonce generates the cryptographic hash. The data in the block is considered signed and forever tied to the nonce and hash unless it is mined.

Miners:Miners create new blocks on the chain through a process called mining.In a blockchain every block has its own unique nonce and hash, but also references the hash of the previous block in the chain, so mining a block isn't easy, especially on large chains.Miners use special software to solve the incredibly complex math problem of finding a nonce that generates an accepted hash. Because the nonce is only 32 bits and the hash is 256, there are roughly four billion possible nonce-hash combinations that must be mined before the right one is found. When that happens miners are said to have found the "golden nonce" and their block is added to the chain.

Nodes:One of the most important concepts in blockchain technology is decentralization. No one computer or organization can own the chain. Instead, it is a distributed ledger via the nodes connected to the chain. Nodes can be any kind of electronic device that maintains copies of the blockchain and keeps the network functioning. Every node has its own copy of the blockchain and the network must algorithmically approve any newly mined block for the chain to be updated, trusted and verified. Since blockchains are transparent, every action in the ledger can be easily checked and viewed. Each participant is given a unique alphanumeric identification number that shows their transactions

In order to prove the honest validation of blocks, there exist numerous mechanisms. The most used ones are the Proof of Work (PoW) and the Proof of Stake (PoS) mechanisms.

Proof of Work (PoW)

In Proof of Work, in order for an actor to be elected as a leader and choose the next block to be added to the blockchain they have to find a solution to a particular mathematical problemGiven that the hash function used is cryptographically secure, the only way to find a solution to that problem is by bruteforce (trying all possible combinations). In other words, probabilistically speaking, the actor who will solve the aforementioned problem first the majority of the time is the one who has access to the most computing powerIt has been widely successful primarily due to its following properties:It is hard to find a solution for that given problem;When given a solution to that problem it is easy to verify that it is correct;Whenever a new block is mined, that miner gets rewarded with some currency (block reward, transaction fees) and thus are incentivized to keep mining. In Proof of Work, other nodes verify the validity of the block by checking that the hash of the data of the block is less than a preset number.Due to the limited supply of computational power, miners are also incentivized not to cheat. Attacking the network would cost a lot because of the high cost of hardware, energy, and potential mining profits missed.

Proof Of Stake(PoS)

In Proof of Work, if Bob has more computational power and energy than Alice — and thus can output more work — he is more likely to win (mine the next block).In Proof of Stake, if Bob has more stake than Alice, he is more likely to win (“mine” the next block).Proof of Stake takes away the energy and computational power requirement of PoW and replaces it with stake. Stake is referred to as an amount of currency that an actor is willing to lock up for a certain amount of time. In return, they get a chance proportional to their stake to be the next leader and select the next block.In PoS the miner of a new block, in this case known as the forger, is chosen in a semi-random, two-part process. The first element to be considered in this selection process is a user’s stake as mentioned aboveThe second most important element is to include a degree of chance to the selection process so as to avoid a scenario where the richest users are always selected to validate transactions, consistently reap the rewards and grow richer and richer.The two most commonly used methods are Randomised Block Selection and Coin Age Selection.

Ethereum

Ethereum is a public blockchain that provides a cryptocurrency called Ether (ETH) (after the fork that happened in July 2017, there exists a version of Ethereum called Ethereum Classic that uses a currency called ETC ), used for paying financial transactions as well as applications processing. Miners replicate, validate, and store data in the blockchain network. Furthermore, they process programs called smart contracts which makes Ethereum a platform for decentralized applications

Smart contract in ethereum

A smart contract is a contract implemented, deployed and executed within an Ethereum environment. Smart contracts are digitization of the legal contracts. Smart contracts are deployed, stored and executed within the Ethereum Virtual machine. Smart contracts can store data. The data stored can be used to record information, fact, associations, balances and any other information needed to implement logic for real world contracts. Smart contracts are very similar to Object oriented classes. A smart contract can call another smart contract just like an Object-oriented object to create and use objects of another class.We can Think of smart contract as a small program consisting of functions. You can create an instance of the contract and invoke functions to view and update contract data along with execution of some logic

IV. Attacking model

In this work, we assume that an attacker or malicious user has a total control over the used network i.e he can selec- tively sniff, drop, replay, reorder, inject, delay, and modify mes- sages arbitrarily with negligible delay. However, the devices can receive unaltered messages. Nonetheless, no assumptions on the rate of the altered messages are made. Besides, the attacker can benefit from a computation power and storage larger than the implemented devices. However, we do not consider physical attacks on devices, where the attacker can retrieve some/all of the object’s secrets such as private keys. We assume that objects are protected against physical attacks

ATTACKS:

1) Sybil attack:The attacker simulates the existence of multiple entities (devices) that send wrong information to the server or management application, in order to elect decisions needed by the attacker.

2) Spoofing attack: in contrast to sybil attack where the at- tacker try to create numerous false or virtual identities, in spoofing attack, the attacker tries to spoof the identity of a legitimate user in order to make use of his privileges.

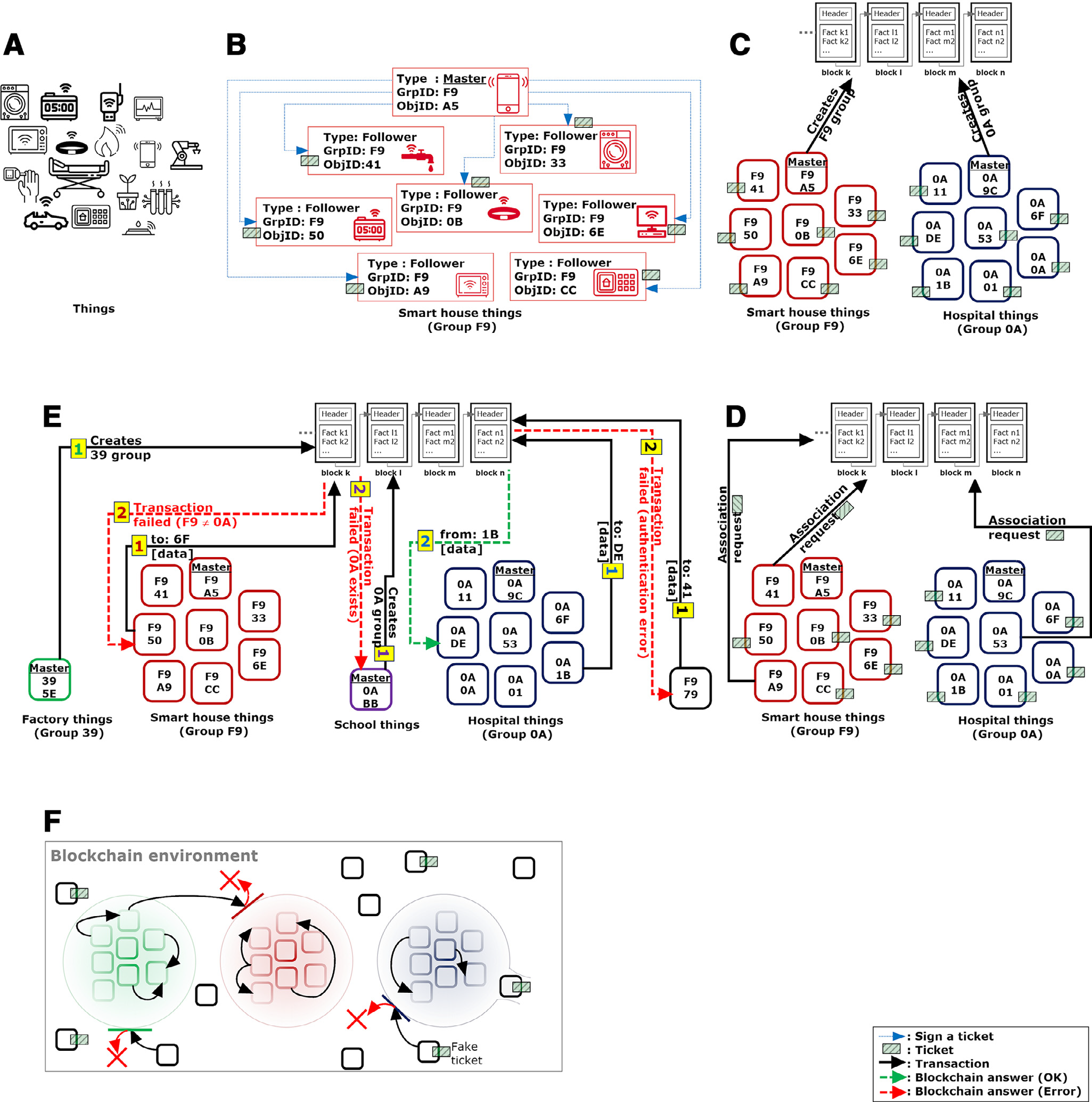
3) Message substitution attack: In a substitution attack, the attacker intercepts valid messages during their transit and alters them in such a way that recipients accept the forged messages as if they had been sent by the original sender.

4) Message replay attack: An attacker can record selectively some messages and replay them without modification at a later time, since successful verification of a message does not certify the correctness of the message’s sending time. In this way, inaccurate information can be intentionally provided to the objects or to the servers. Message replay attack is usually combined to a message removal attack.

5)DDoS:Distributed denial of service attacks is characterized by the explicit attempt by attacker to prevent the legitimate use of a service

# Proposed Methodology

The main goal of the approach[1] is to create secure virtual zones so that the devices within this virtual zone considers themselves as neighbours hence non malicious and only talks with each other only.They don't talk to other devices outside this bubble.The communication within this bubble is considered as transaction and must be validated by blockchain in order to be seen as valid



1. Initialization phase.

(step B & C in above fig)

In this phase a device is assigned as a Master of the bubble (It owns a private/public key-pair), which can be considered similar to a certification authority but local to a group of devices.Any given device can be the Master.After creation of group identifier and master id it sends them to blockchain,where it get verified for its uniqueness. Besides, each object that makes up part inside the bubble is called a Follower. Each Follower generates an Elliptic Curve (EC)(or RSA) private/public key-pair.

Then, each Follower is provided by a structure called a ticket, which represents a lightweight certificate of 64 bytes that contains: (1) a groupID , which represents the bubble that the object will be part of, (2) an objectID , which represents the Follower’s identifier in the bubble, (3) pubAddr, which represents the Follower’s public address(4) a Signature structure which represents the Elliptic Curve Digital Signature Algorithm (ECDSA) signature using the private key of the bubble ’s Master The ticket structure is as follows:



The following operations are described(step D):

1.The first client’s transaction represents an association request. The sent message is signed with the Follower’s private key and contains the Follower’s ticket.

2.When the blockchain receives the transaction it verifies its integrity by verifying the signature with the Follower’s public key. Then, the Follower’s ticket is verified using the Master’s public key, since it represents the entity that signed it.

3.If the ticket is valid, then, the blockchain stores an association of its grpID, objID and the public key. Thus, it stores (XX, YY and PubKey\_F).

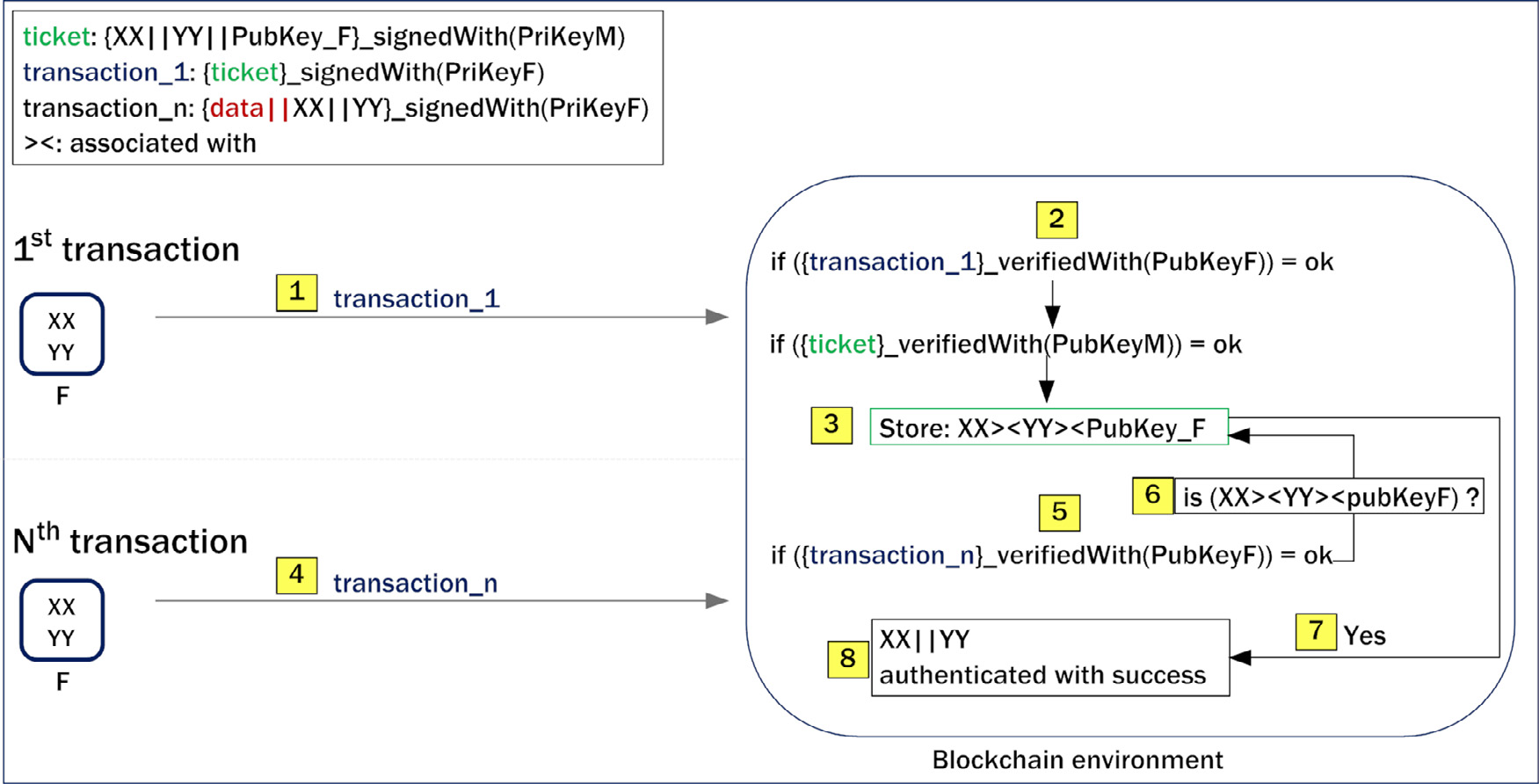
4.The fourth step describes the case where F sends another transaction (transaction n) than the association request. This transaction contains: (1) the exchanged data, (2) XX, (3) YY and (4) the ECDSA signature of the concatenation of the previous fields using the Follower’s private key.

5.When the blockchain receives the transaction it verifies its integrity by verifying the signature with the Follower’s public key.

6.If the signature is valid, the blockchain verifies if the public key used for the transaction’s verification is stored and associated to the grpID and objID sent within the transaction.

7.If the association is stored and is valid then the device is authenticated with success.

“Once the first transaction (association request) of a Follower is successful, the latter does no longer need to use its ticket to authenticate itself (sends it within the exchanged messages).”



Requirements:

* Flask framework in the node running the blockchain
* Crypto library of python
* Tkinter for GUI of blockchain.(Not implementing on IoT devices as of now)
* Request for handling the get requests.

Modules:

Node :

* Methods use to create master and follower as well as generates there keys and corresponding tickets for the concerned
* Method to sign the ticket of follower
* Methods to verify the ticket of follower
* Method to submit the info

Peer :

Contains Blockchain class:

* Methods for creating genesis block
* Methods for adding the block
* Methods for proof of work
* Methods for adding new transaction
* Methods for validating the proof
* Method for checking chain validity
* Method for checking existence of contracts(groupid,object id already exist or not)
* Methods for for achieving consensus among nodes

# Work done

The above-proposed solution has been proposed by the [1] and the working of it is unaltered in our implementation; all the security goals and other requirements and proposed benefits are an integral part Of our implementation as well. Our implementation takes the task of implementing the backend blockchain stack, the smart contract and the transaction infrastructure from scratch keeping in mind the IoT framework.We have been successful in implementing the blockchain base as of now. We have been able to do dummy transactions on our system till now. We have developed the nodes and integrated the blockchain with the Flask framework to achieve the get requests from the node so as to facilitate transaction requests from the nodes. This is GitHub link of our code: <https://github.com/vibhor70/bubbles-of-trust>.

The following are the methods used in our implementation:

**class blockchain**- It defines all the variables and the functions of the class blockchain.

**create\_block**- It creates a new block in the blockchain.

**get\_lastblock**- It returns the last block.

**proof\_of\_work**- It is the methodecial puzzle to be solved by the miner to add a block.

**hash**-It returns the hash of the object.

**is\_chain\_valid**- It is used to check the validity of the chain.

**check\_master**- It checks the uniqueness of the master and the group id called before addition of new master.

**check\_follower**- It checks for the uniqueness of the follower in the group.

**check\_message**- It checks the groupid of the recipient and the sender.

**add\_transaction\_master**- It adds master and creates its bubble.

**add\_transaction\_follower**- It adds the follower to a bubble.

**add\_transaction message**- It adds a message to the blockchain as part of valid conversation.

**add\_node**- It adds the node to the blockchain.

**replace\_chain**- It selects the longest chain among all the nodes.

**generatekey**- It generates a pair of ECC public-private key.

**generate\_ticket**- It generates a ticket with a signature signed by the master of the group.

**verify \_ticket**-authenticates the ticket with the private key of master

**mine\_block**-For mining of transaction blocks

**get\_chain**-return the whole blockchain as well the length

**add\_transaction**-use for adding transaction of three types -Master,Follower,Messages

every transaction has particular structure and is checked for its uniqueness as well as its authenticity,and only after that is added into the temporary pool of transactions

**connect\_node**-used to keep updated list of all the connected nodes

**replace\_chain**-finds max chain length is all the blockchains of all the nodes and replaces with the longest chain

**generate\_key**-use for creation of public private keys

**get\_ticket**-For generation of followers ticket

{

"GroupId": "101",

"ObjectId": "object",

"PubAddr":

"-----BEGIN PUBLIC KEY-----\nMFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEFwz5g4t17HzkQvvNa23hBQNiR8Ix\nO1Q0Pr98qcybUs1TF3UdiFdCIfCfS6GSQOEAF3NpYgotsGA+70zyuKphLg==\n-----END PUBLIC KEY-----",

"Signature": "dcYnj/81Bmb0HokZJrJ+PZXc3FZJKHV71qXK48HLWHMSKIRhK1P9ahb+znLO4uOMCbsn/glFEC4LAJCHRpB3qQ=="

}

# Conclusions

The importance of security of IoT devices is of paramount importance and we believe that blockchain will lead it in future due to the sheer number of devices on the internet today~7billion.The proposed approach in [1] will have a tremendous impact on security of devices in every aspect.

##### References

1. Mohamed Tahar Hammi , Badis Hammi , Patrick Bellot, Ahmed Serhrouchni, Bubbles of Trust: A decentralized blockchain-based authentication system for IoT,20 June 2018
2. Christidis K ,Devetsikiotis M . Blockchains and smart contracts for the internet of things. IEEE Access 2016;4:2292–303 .
3. Malviya H. How blockchain will defend iot 2016.
4. Huh S ,Cho S ,Kim S . Managing iot devices using blockchain platform. Proceedings of the 2017 19th international conference on advanced communication technology, ICACT. IEEE; 2017. p. 464–7 .
5. Bahga A ,Madisetti VK . Blockchain platform for industrial internet of things. J Softw Eng Appl 2016;9(10):533 .
6. Ruta M ,Scioscia F ,Ieva S ,Capurso G ,Di Sciascio E . Semantic blockchain to improve scalability in the internet of things. Open J Internet Things 2017;3(1):46–61 .
7. Dorri A ,Kanhere SS ,Jurdak R . Blockchain in internet of things: challenges and solutions. arXiv preprint arXiv:160805187 2016 .Dorri A ,Kanhere SS ,Jurdak R ,Gauravaram P . Blockchain for iot security and privacy: the case study of a smart home. Proceedings of the 2017 IEEE international conference on pervasive computing and communications workshops, PerCom Workshops. IEEE; 2017. p. 618–23
8. Ouaddah A ,Abou Elkalam A ,Ait Ouahman A . Fairaccess: a new blockchain-based access control framework for the internet of things. Secur Commun Netw 2016;9(18):5943–64 .Ouaddah A ,Elkalam AA ,Ouahman AA . Towards a novel privacy-preserving access control model based on blockchain technology in iot. Europe and MENA cooperation advances in information and communication technologies. Springer; 2017. p. 523–33 .