SECURE AUTHENTICATION IN IOT USING DYNAMIC KERBEROS

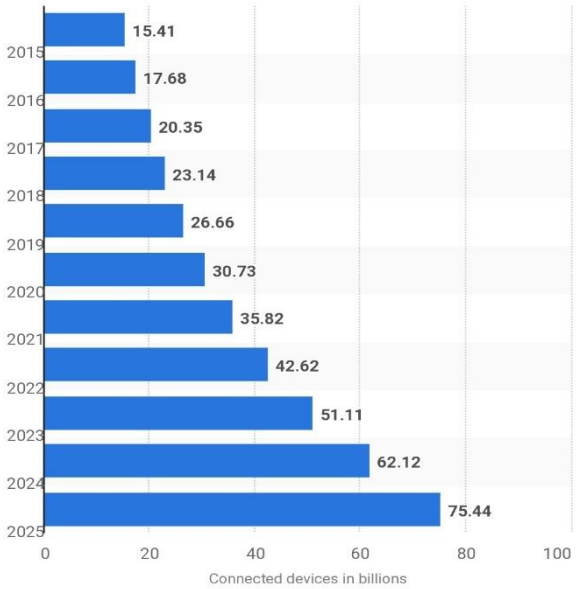
[1]Gagandeep Singh, [2]Dr. Suresh Kallam,
[1] M.Tech Computer Science and Engineering, [2]Professor, Computer Science and Engineering,
[1]School of Computing Science and Engineering, [2]School of Computing Science and Engineering
[1]Galgotias University, Greater Noida, India, [2]Galgotias University, Greater Noida, India

Abstract : IoT technology and its devices are emerging at a huge scale. A lot of sensitive data is transferred over networks in IoT, which could be either secure or insecure. While transferring such data over the network, it is more likely that attacks like data theft, manipulation of server, device and data will surface. Thus, we require an efficient and secure protocol to protect the user privacy and sensitive data. The general implementation Kerberos is prone to attacks because the major components i.e. Authentication Server (AS) and Ticket Granting Server (TGS) can be easily attacked due to their fixed positions and attacker can also monitor the conversation between client and AS due to the presence of single layer of communication channel. So, in this paper we are proposing an efficient and effective authentication scheme by making some changes to the existing Kerberos, the AS and TGS are made dynamic that means instead of relying on fixed nodes, multiple nodes can participate to become AS and TGS along with the help of fog computing and Smart Contract.

***Index Terms* – fog nodes, smart contract, hash, authentication server, ticket granting server**

I. INTRODUCTION

Internet of Things. It is a group of interconnected devices (smartwatches, mobiles, cameras, cars, E appliances, smart home equipment: lights, speakers etc.) which can be accessed and data can be exchanged over the internet. The growth of IoT devices is at peak. It took just one year for IoT devices to reach from 5-6 millions to billions with a possibility of 25-30 billion before 2020 and 75 billion by 2025 (according to Statista) [7].



IoT devices facilitates a large scope in the areas where huge systems and computers are inefficient to use. They are simple devices with limited set of functionalities and computational strength. IoT devices have to handle a huge amount of confidential and sensitive data yet they are not capable of defending and protecting themselves from attacks and hacks and they are resource limited devices. Therefore developers have to adopt some counter measures to make IoT devices attack proof. The two major things to provide while securing IoT devices are IoT encryption and authentication. IoT encryption refers to securing the data being transferred over the internet from hackers and attackers by any kind of encryption technique. IoT authentication refers to the authentication of users accessing the IoT devices which can be either as simple as a password or as complex as biometrics. It should be kept in mind that resources in IoT devices are limited and hence they will not be able to perform huge computation.

Fog computing is an efficient way to make a different but capable system to carry out heavy workload on the behalf of several IoT devices. A single system in a fog computing environment acts like a guardian and is capable of performing computation, storing data for a group of IoT devices. All the requests and response regarding those IoT devices are handled by the fog computer.

To implement secure authentication, Kerberos protocol is being used. It provides authentication for server and client in a strong manner. The Kerberos protocol makes use of tickets to grant access to various services. The tickets are obtained by the Authentication server by the client using a specific key which was generated using the password of the client. The basic version of Kerberos protocol is centralized and has fixed Authentication Server, which makes it more vulnerable to attack as there is a single point of failure. To make it decentralized, the position of the Authentication server has to be changed i.e. the role of the authentication server has to be transferred to another device. Therefore making it decentralized will help in the reduction of attacks to a large extent.

In this paper, the aim is to make Kerberos system work in a more dynamic fashion so as to provide robust authentication in IoT environment. This will be achieved by making the roles of Authentication Server and Ticket Granting Server not limited to a single node. This will be done with the use of a Smart Contract which randomly assigns the functionalities of AS and TGS to different nodes after a particular amount of time.

II. Related Work

Authentication is something which is used to find out two things: who the user is and is he/she the one he is pretending to be? This can be done with the help of a simple plain text password as in PAP i.e. Password Authentication Protocol or using a more complex hashed password as in CHAP i.e. Challenge Handshake Authentication Protocol [1]. But in both the cases, password is being transferred over the network which makes it prone to password guessing attacks like Brute Force.

In [2], a famous method which is being used for authentication in IoT devices is Open Authorization also known as OAuth. It makes use of tokens to grant and revoke access to the resources. There are two major components in OAuth – Authorization Server, which is responsible to authorize the user and generate tokens to access the resources and secondly, the user itself. Once the user is authorized, it can then access the resource server which only grants services if the user has valid token.

In [3], a 3 tier Kerberos system is presented to authenticate the users in a smart home system. The three level are – Level 1: The user login, Level 2: the authentication and Level 3, the distribution of tickets and services. A simple traditional Kerberos is being used here which has a single point of failure i.e. Authentication Server. If the AS gets compromised, the whole system will fail. Similarly in [4], a simpler version of Kerberos is being used by merging the Authentication Server and the Ticket Granting Server. A single server manages both the authentication and ticket granting tasks, which makes it even more prone to attacks as now there is one point to attacks and you get hold of all the functionalities of the Kerberos server.

Moreover, the work in [5] depicts how the Kerberos system fails in a real environment. Apart from being centralized, it is also prone to attacks like Replay attacks where the attacker captures the packet from the network and sends it to the server pretending to be the user. In [6], a decentralized system is being used to prevent single point of failure and hence the block chain based authentication is used to make it less prone to attacks. Smart Contracts are used in the system to act as a governing body in the system. It decides which user should access what. The communication is done based on the tokens.

III. Proposed Model

3.1 Design

In this section, we will define a system for Kerberos based authentication using Fog nodes. The system will have *n* nodes serving as IoT devices. As these nodes are IoT devices, they will have a limited computing power as well as limited resources such as storage. Therefore, all the nodes have one Fog node as their parent as shown in Fig. 1. These fog nodes will perform all the necessary computation tasks on behalf of those IoT devices. All these fog nodes capable of being Authentication Server as well as Key Distribution Centre, which are the two most important pillars in any Kerberos system.

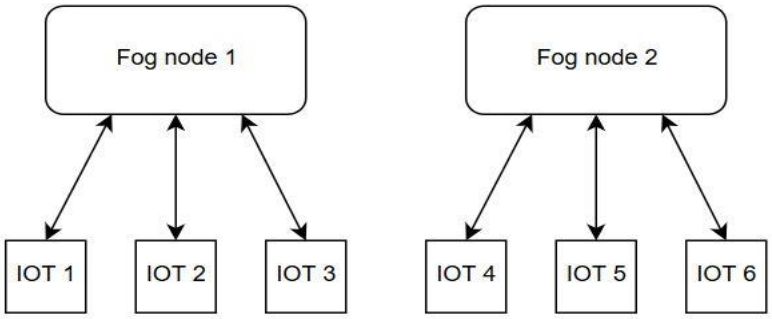


Figure 1

To make Kerberos act as a decentralized system, the fog nodes acting as AS and TGS will be changing their position after certain point in time.

Example- if fog nodes *f1* and *f2* are AS and TGS respectively, then after time *t* fog nodes *f3* and *f4* will be AS and TGS.

This switching happens on a random basis so no one can guess what fog nodes will serve as AS and TGS next. To perform and govern this switching, a Smart Contract is used. Smart contract, as the name suggests, is a kind of contract which has a set of rules and regulations to be followed in a particular system. The architecture of the system is given in Fig. 2

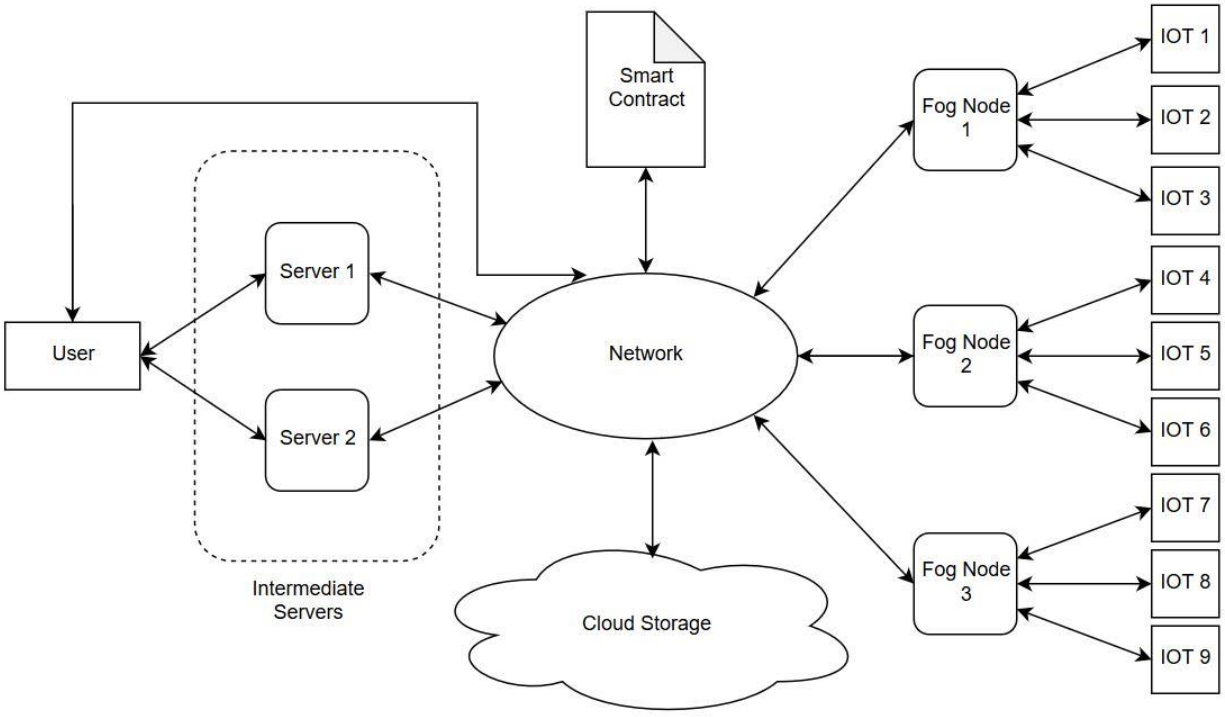


Figure 2

Now, there is a single channel of communication between the user and the Authentication Server. As per the Kerberos protocol, no password will be transferred using the communication channel, but still the single channel makes it much easier to attack. Therefore, a two tier authentication is used in our system so as to make it a bit harder to attack. What a two tier authentication means is that between user and AS, there exists two intermediate servers/nodes. The goal of these intermediate servers is to forward the data received from the user to the AS.
But there's how the actual work will be performed –

- First the client will generate a random number on its side.
- Client will then concatenate the random number with its username and finds the hash of it.
- Also, the client sends its user name and random number to the AS encrypted using the public key of the AS.
- The generated hash will be divided in two parts and then sent to each intermediate servers encrypted using the public key of the intermediate servers so that only they can access it.
- After getting the hash from the client, both the servers will eventually forward it to the AS.
- The AS will decrypt the message received from the client, find the hash code of the concatenated username and random number, and then will compare it with the hash received from the intermediate servers.
- If both are same, secure authentication is established the process of Kerberos will be continued normally.

The different entities in system are –

- User – Users are the ones who issue requests to access a particular IoT devices. As soon as the user is authenticated, it is then granted access to communicate with the certain IoT device.
- Two tier authentication – It consists of two intermediate nodes which act as a bridge between the user and fog node acting as the Authentication Server.

- Cloud – Cloud acts as repository for all the IoT related data. The can provide as well as store data coming from the IoT devices. In other words, it acts as a Big Data store.

- Fog node – A fog node is responsible for the computation, storage and other stuff on behalf of the IoT devices. This is done because of the resource constraint property of IoT devices. Because of the fog nodes the IoT devices are relieved of any huge task which will affect is performance.

- Smart Contract – To govern all the changes made dynamically in the system a smart contract is used. The smart contract is responsible for mapping roles of AS and TGS to the fog nodes, as well as mapping of fog nodes to IoT devices. All registration, authentication, access control functionalities are governed in a decentralized manner through the smart contract.

3.1 Working

The system will work as follows –
The user will issue a request to access any particular IoT device. The request goes to the fog node acting as the authentication server with the help of smart contract. Suppose at time *t=1*, node *f1* is AS and node *f2* is TGS. The request will go to the node *f1*, and hence the process of Kerberos authentication begins. The request will go in the form of username concatenated with the random number which goes straight to the AS, as well as passes through the two intermediate servers. As soon as the request is verified, the AS sends a ticket T_{AS-U} to user.
Now the user has the ticket to the TGS, it will now send another request to the TGS demanding access to the particular IoT node. The request will contain T_{U-TGS} on the TGS end, the request is verified and ticket to access the IoT node T_{TGS-U} is provided. And hence, the user will be able to communicate with the IoT device.
At time t=1+x, where x is some predefined amount of time, the Smart contract select two random nodes to act as AS and TGS. This is done to minimize the attack on the system. After the roles of AS and TGS are changed, the old nodes *f1* and *f2* will no longer receive requests from the user. The complete exchange of information is shown in Fig. 3

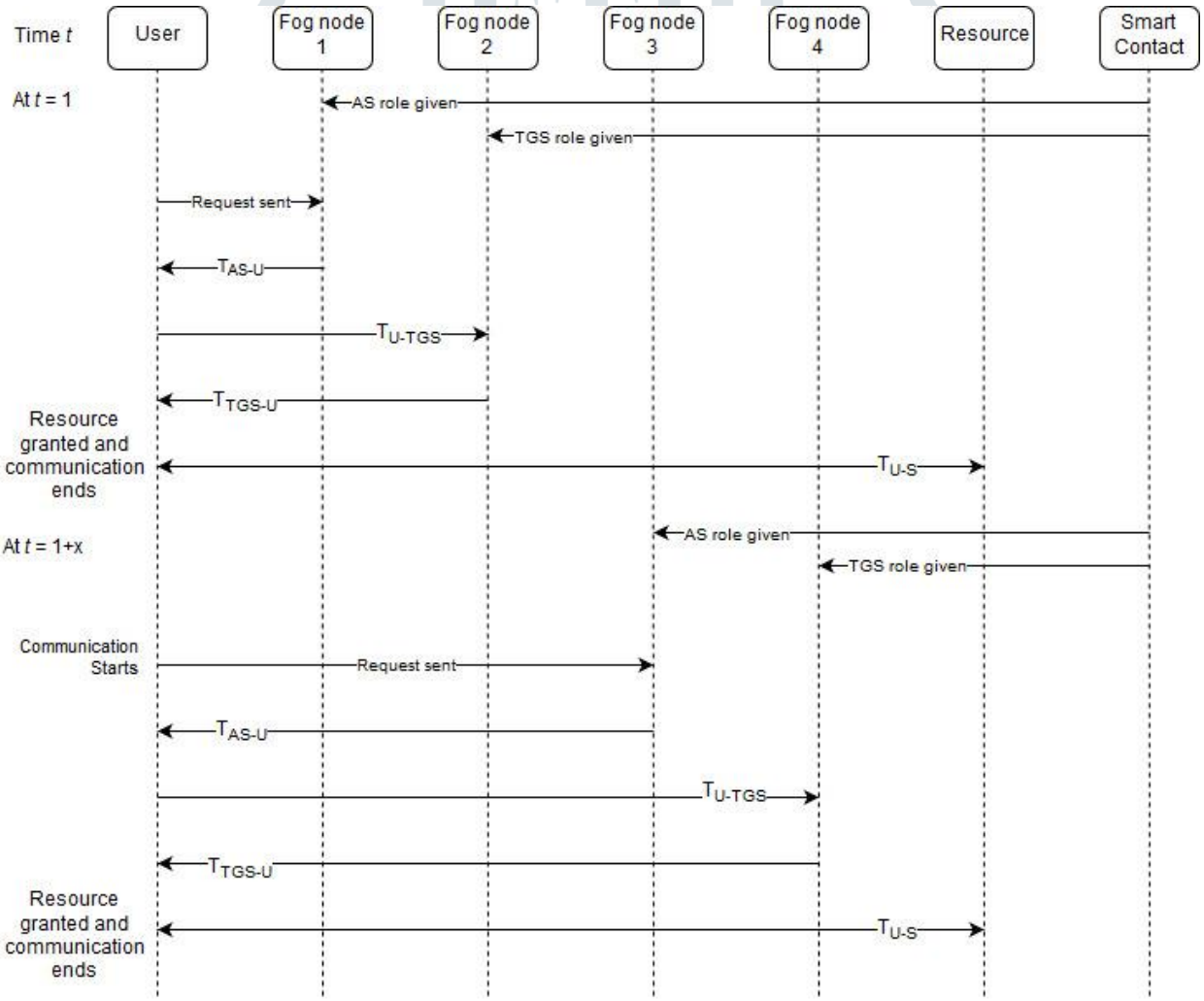


Figure 3

IV. ALGORITHM AND RESULTS

4.1 Algorithm

Assume the system to be K, which consists F_x fog nodes, N_x users, I_x IoT devices, two intermediate servers IS_1 and IS_2, where x ranges from 1 to n

Below are the algorithmic steps of what and how each entity will perform,

Admin:

- Add users N to the system
- Add Fog nodes F to the system
- Add devices I to the system
- Add another admins to the system

User:

- Username gets combined with a random number
- The whole string gets converted into a hash H
- H is divided in two halves H1 and H2, each part is sent to each intermediate server
- The original hash H is sent to the AS
- Waits till the response of AS
- Receives T_{AS-U} containing TGT and a S_{AS} session key
- T_{AS-U} gets decrypted with own password
- Makes a ticket/packet T_{U-TGS} containing TGT and Authenticator which is encrypted by using S_A
- Sends T_{U-TGS} to TGS
- Receives T_{TGS-U} containing Service Ticket and S_{TGS} session key
- T_{TGS-U} gets decrypted using S_{AS}
- Makes T_{U-S} containing Service Ticket and Authenticator which is encrypted using S_{TGS}
- Sends T_{U-S} to Server
- Waits for Server's response

Intermediate Server:

- Receives hashed text containing a part of concatenated username with random number
- Sends it to the AS

Authentication Server:

- Receives hash H user and H1 and H2 from intermediated servers
- Joins H1 and H2 to form H_x
- Compares both hashes H and H_x
- Creates T_{AS-U} containing TGT (Ticket Granting Ticket) encrypted by private key of TGS and a S_{AS} session key
- Encrypts T_{AS-U} with user's password and sends it to user
- Waits for new request

Ticket Granting Server:

- Receives T_{U-TGS} from user
- Decrypts T_{U-TGS} using S_{AS}, checks TGT and Authenticator
- Creates T_{TGS-U} containing Service Ticket encrypted by private key of Server and a S_{TGS} session key
- Encrypts T_{TGS-U} with S_{AS} and sends it to user
- Waits for new request

Smart Contract:

- At time t, choose any two random fog nodes x and y
- Make x as AS and y as TGS
- After time t+n, choose any two random fog nodes s and t
- if s = x or t = y, choose again
    - if s = t, choose again
- After time t+2n, choose any two random fog nodes i and j
- if i = s or j = t, choose again
    - if i = j, choose again
- Repeat these steps until system stops

4.2 Results

By using intermediate servers between client and main server, increase the no. of paths which carries the user information. A single path can easily be intercepted, but as the number of path increases, it becomes more difficult. Fig. 4 shows how an increase in no. of paths can lead to less chances of user information being intercepted.

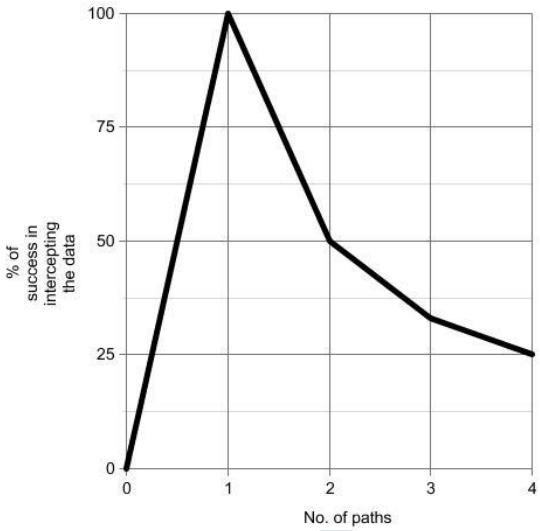


Figure 4

The main goal of this system to increase security. With so many nodes to choose from, the attacker will have a hard time finding the exact place to attack. Fig. 5 shows how the attacker has an upper hand when it comes to attack a simple centralized Kerberos system as security factor is constant and small. The increase in the number of nodes does not increase or decrease the security of the system. Security definitely increases by making the system act as a decentralized one. Hence, Fig. 6 shows how the security becomes directly proportional to the number of nodes and increases gradually.

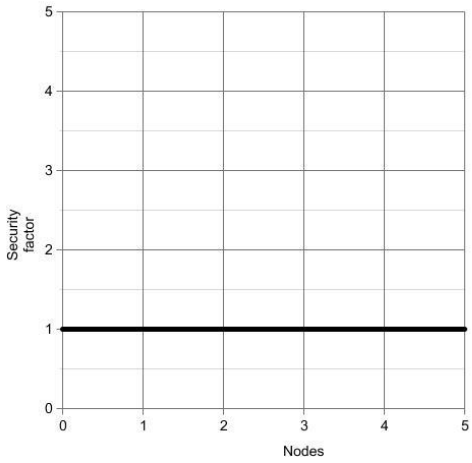


Figure 5

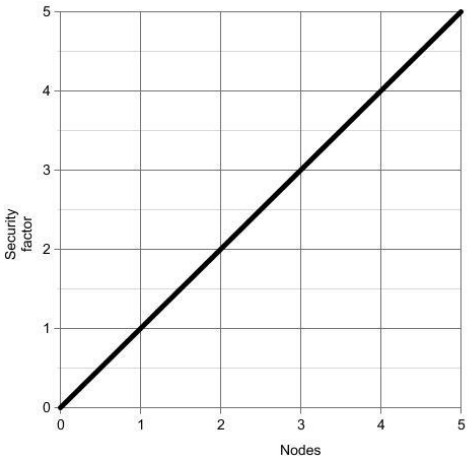


Figure 6

Fig. 8 shows how likely the system is prone to attacks when using traditional approach. Whereas, Fig. 9 shows how that probability goes down when number of nodes rise up.

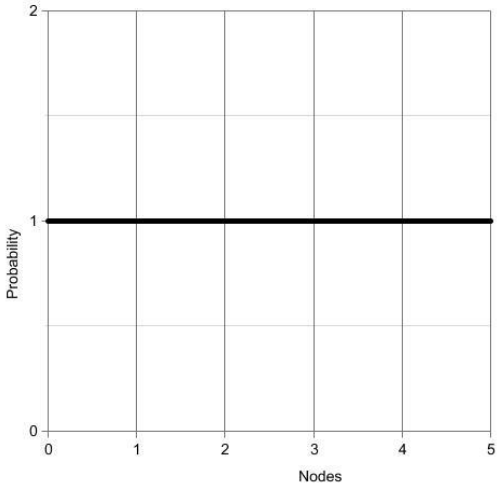


Figure 7

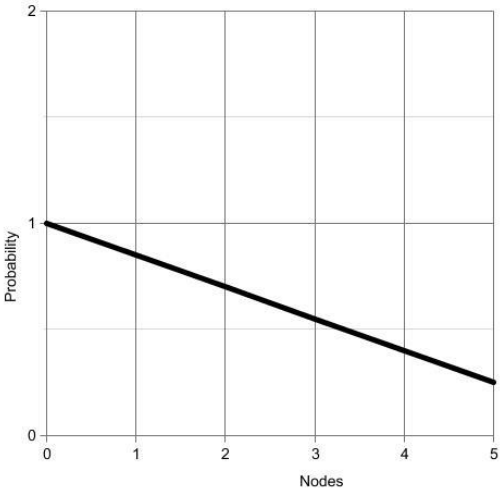


Figure 8

Hence, here is a head on comparison between the two systems

Table 1 Comparison between old and new approach

System	Old System	Proposed System
Working Manner	Centralized	Decentralized
Governing Unit	Key Distribution Centre	Smart Contract
Can the governing unit be attacked?	Yes, as it an actual system and its position is fixed	No, as it is just bunch of code running on a virtual computer.
Probability of attacked node being either AS or TGS	1	<1
Protection from attacks such as replay attacks	No	Yes

V. CONCLUSION

In this paper, we have distributed the roles Authentication Server and Ticket Granting Server of the traditional Kerberos system with the help of Smart Contract. Since the attacker will find it difficult to find locations of AS and TGS, the chances of system getting compromised gradually decreases. And if somehow attacker finds AS or TGS, their roles will eventually be passed on to some random nodes in the network. Also there exists a two tier communication between user and AS so as to prevent attacks like replay attacks and eavesdropping, which makes the system more secure than ever before.

REFERENCES

[1] S. Z. Syed Idrus, E. Cherrier, C. Rosenberger, and J.-J. Schwartzmann, "A review on authentication methods," vol. 7, pp. 95–107, 06 2013.
[2] S. Cirani, M. Picone, P. Gonizzi, L. Veltri, and G. Ferrari, "Iot-oas: An oauth-based authorization service architecture for secure services in iot scenarios," IEEE Sensors Journal, vol. 15, no. 2, pp. 1224–1234, Feb 2015.
[3] P. P. Gaikwad, J. P. Gabhane and S. S. Golait, "3-level secure Kerberos authentication for Smart Home Systems using IoT,", 2015 1st International Conference on Next Generation Computing Technologies (NGCT), Dehradun, 2015
[4] Jayati Ghosh Dastidar, "An Authentication Protocol based on Kerberos", Int. Journal of Engineering Research and Application, Vol. 7, Issue 7, (Part -4) July 2017
[5] Nazri Abdullah, Anne Håkansson, Esmiralda Moradian, "Blockchain based Approach to Enhance Big Data Authentication in Distributed Environment", Ninth International Conference on Ubiquitous and Future Networks (ICUFN), IEEE, 2017
[6] Randa Almadhoun, Maha Kadadha, Maya Alhemeiri, Maryam Alshehhi, Khaled Salah, "A User Authentication Scheme of IoT Devices using Blockchain-enabled Fog Nodes", 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA),2018
[7] https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/