Advances and Applications in Mathematical Sciences
Volume 20, Issue 12, 2021, Pages 3219-3233

LINEAMENTS AND RESULTANT CONSEQUENCES, POTENTIAL SOLVENT OF LATENCY AND CONSUMPTION OF ENERGY IN IOT

KESHETTI SREEKALA

Department of Computer
Science and Engineering
Mahatma Gandhi Institute of Technology
Gandipet Hyderabad-500075, India
E-mail: ksrikala_cse@mgit.ac.in

Abstract

There is no doubt that IoT made our lives easy. With the ease of use and comfort of living the usage of IoT is growing but at the same time the numbers of possible threats are also growing. This paper discusses the Lineaments and the resultant consequences , the possible threats, solutions and issues in IoT that are yet to be solved along with a potential solvent for Latency and Consumption of Energy in IoT. The lineaments like correlation, variety, captivity, unaccountability, inattentiveness, familiarity, mobility, omnipresence of IoT devices and its applications are different from the custom computer and Internet applications. We propose a solution based on fog computing which can solve few of the above mentioned problems. In order to design systems that provide Latency support and low energy consumption it is important to know about these lineaments. These lineaments also help in (1) Identifying possible attacks, vulnerabilities and leakage of data (2). The new researchers to do progressive work towards IoT Security, seclusion, Energy Consumption and Latency Support.

Introduction

As per the Statistics given in the website of safeatlast [1] it is recognized that 127 new devices are getting connected to the Internet for every second and at present there are 26.66 billions of IoT active devices. It is estimated that by 2025 there will be 75 billion devices in the World. It is also estimated that 40% of the IoT devices are going to be used in healthcare industry. Along the line of advancement of IoT devices the possible attacks are also advancing

2010 Mathematics Subject Classification: 68.
Keywords: attack, Fog computing, lineaments, threat.
Received November 16, 2020; Accepted December 20, 2020

these days. I feel that Low energy consumption and lower latency time are important for use of IoT devices. The numbers of IoT users are growing day by day but still the energy consumption remains an open issue. Already so much of survey has been done in the context of latency time and energy consumption of IoT devices. Lin et al. [2] and Li et al. [3] explored about different types of attacks and challenges through the layers. Sicari et al. [4] discussed about the continuous ongoing challenges and the anticipating solutions focus at different levels of security like authenticity, controlling access, seclusion and assurance. Yang et al. [5] and Trape et al. [6] discussed about pertinent limitations of IoT devices like restricted processing power and charging capacity. But there are many more IoT constraints that could affect the latency time and energy consumption of IoT devices. Fu et al. [7] discussed about opportunities and possible risks with respect to hospital and home which are the two different applications. Roman et al. [8] presented promising findings at different levels of security. The sequence of activities or the summary of this paper is

(1) In order to identify the possible IoT threats and the causes we first discuss the lineaments of IoT.

(2) To understand these lineaments in a better way we also discuss the possible threats, challenges associated with research and opportunities with respect to each and every lineament.

(3) We present a fog computing based solution for maintaining the lower latency time and less energy consumption of IoT.

The rest of the paper is organized as follows. The following sections of this paper describe about Lineaments of IoT and Associated terminology, background of work, detailed idea of the algorithm proposed to reduce latency and energy consumption, implementation part, results followed by conclusion and future work.

Lineaments of IoT and Associated Terminology of Related Work

It is identified that there are eight important lineaments in IoT where more focus is required in order to reduce the threats/attacks. They are shown in figure 1 and are termed as Correlation, Variety, Captivity,

Unaccountability, inattentiveness, familiarity, mobility and Omni presence.

Correlation: Apart from communicating with each other like custom computers IoT devices could also be controlled by many other devices and external environmental conditions using the services like If This Then That (IFTTT). For example if a thermometer detects a raise in room temperature and the air conditioner that is connected to a smart plug which is in off state then the windows of the home are going to be opened automatically as shown in the figure 2. In this example the smart plug is connected to the public network so the attacker might not be targeting at the thermometer or the smart window in order to open the window. Instead the attacker can make the status of the smart plug to off, as a result the temperature of the room raises and the windows will get opened. This result in a threat called physical security breakage. Because of the public network connection this problem is in existence. This is called over privilege [9] problem and is commonly found in today's IoT applications. In order to solve the Over privilege problem Yunhan et al. [10] proposed a system which is based on context permission called Contex IoT but this takes more data from run time and data and control flow as it interprets the correlation of IoT devices at an early stage. Hence more effective solutions are required to address the problem caused by correlation.

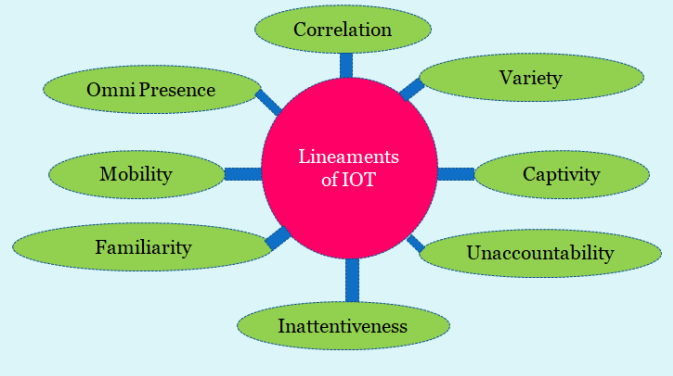

Figure 1. Lineaments of IoT.

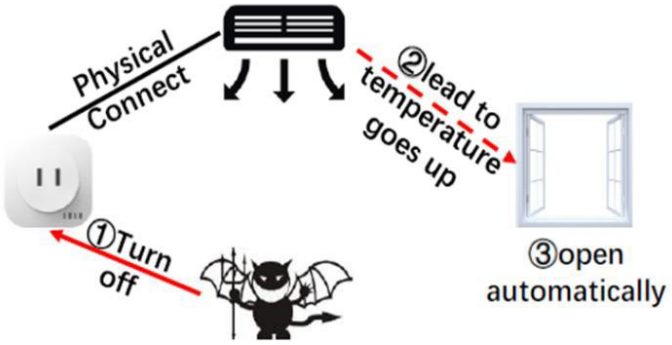

Figure 2. Physical attack because of Correlation.

Variety: IoT devices interact with physical environment and many of the IoT devices are specifically designed to perform a single task. IoT device might run on a single chip with few bits of RAM or Flash. On the other hand a complex machine tool might take more memory and these two different scenarios may use different protocols for communication. There are many different authentication, communication and wireless access protocols available in the market. This is known as variety. For example one possible attack was found by Liu et al. [11] which resulted with the use of Joy Link protocol of JD. Chen et al. [12] developed a framework for systems that are based on Linux. Hence a more suitable model for intrusion detection and prevention is needed.

Captivity: Most of the IoT devices used in industry and medical field are very small in size. Thus they fall under the category of light weight devices. These devices have less storage and less processing power. On the other hand these devices have to work continuously for long time in environments like agriculture fields, military and real time processes. At places like agriculture fields the many not be facility to charge the device. Hence processing power, storage capacity, Latency of IoT and power supply are some of the restrictions on IoT devices. Working under restricted environment is called captivity. ARMor [13] which is a lightweight software fault isolation that can be used for the application code developed in a sandbox but it causes high performance overhead for the programs which needs address checking many times. Koeberl et al. [14] demonstrated a complete set of functions for light weight devices which can provide a trusted execution. But this requires to change in the underlying hardware architecture of MCU. EPOXY [15] and

MINION [16] addresses above mentioned problems but their protection scheme works on static analysis of source code as a result it causes burden on the developers. Hence secure light weight algorithms are in need but lightweight algorithms are not as much secure as the conventional cryptographic algorithms.

Unaccountability: As the data generated is enormous this feature is named as Unaccountability. In the recent history, Mirai botnet made many of the IoT devices to compromise. Not only this botnet, but also other kind of botnets like IoTroop [17] are getting generated day by day. As a result the issue is with the protection of IoT devices and the generated data. These botnets also leading to DDoS [18] (Distributed Denial of Service) attacks. Most of the IoT applications are industry oriented and this kind of attacks are not basically on a website instead they target on the social security of an individual or industry. Hence there is an emerging need for researchers to develop antivirus software which can detect this kind of attacks and preserve security of IoT devices. Zhang and Green [19] developed a lightweight algorithm by considering IoT devices and their surroundings. This algorithm can differentiate between legitimate and malicious requests, but their assumption is sending similar kind of text in every attack which was not practically correct assumption. The existing detection attacks for DDoS are applicable only for 6LoWPAN [20] and Smart Grid [21].

Inattentiveness:

IMD devices (Implantable medical devices), Industrial IoT devices and the devices that are used in military and agriculture purpose are supposed to work for longer hours without even charging them. Since the IoT devices are untouched for a longer time this feature is named as Inattentiveness. Hence Reliable execution environment is to be built by the researchers and scientists. Trust Shadow [22] is one such kind of execution environment created for ARM Cortex-A processor but this does not include any provision to support light weight processors. Familiarity: The implantable and wearable IoT devices are collecting our biological data like blood pressure, heart beat rate and so on. They are not only collecting this data but also checking our surroundings to monitor temperature kind of things. This made relation of humans and IoT devices intimate. Hence this feature is named as

Familiarity. Attackers can detect and attack by using sensors to notice smoke and/or carbondioxide [23] released from the room. Smoke and carbondioxide are some features to find number of persons in the room. Though there are some techniques like smartgrid [24] and privacy data sharing with the cloud service [25] they are causing delay. Hence more study is to be done on data gathering, transfer, usage, stockage and sharing. Mobility: The wearable devices as well as smart vehicles are moving from one network to another network because of the persons are moving from place to place. The movement of IoT devices from one place to another is known as Mobility. The probability of occurring such kind of attacks can be reduced by changing the configuration [26] of the IoT devices dynamically. Omnipresence: Day by day use of IoT devices is rapidly increasing. This all lead to a time where humans are more dependent on smart devices. As these devices are inseparable from humans it is known as Omnipresence. The consumers must be aware of the importance of data that is being generated and shared and should not use the default passwords which make the attackers job easy. Manufacturers must also produce the devices with basic security. Operators [27] think that attackers do not know how to operate them and do not attack on these devices but this is not the case in reality. Background Work The basic model for IoT devices has basically three layers. These layers are shown in figure 3 namely terminal layer, fog layer and cloud layer. As shown in the below figure fog layer sits between terminal layer and cloud layer. Different terminals present in the terminal layer can establish the connection and make communication with a node that is present in the fog layer. This node acts as a local cloud node and can do all the computations required by the terminal nodes as a result data transmission delays are reduced. As the numbers of terminals that are getting connected to a fog node are increased the fog node loses its energy. The existing work has been studied and a compact description of this is tabulated in table 1. Most of the existing models did not considered the energy consumption of the fog nodes and / or reliability and robustness of the fog nodes. This made us to study and develop the proposed model. We can say that this model can better provide robustness and reliability when compared with the existing models.

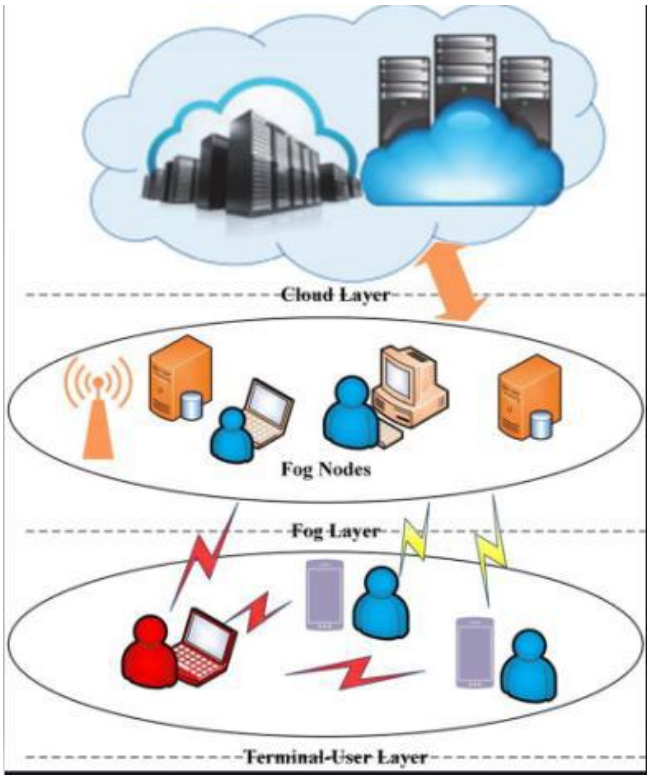

Figure 3. The basic three layer.

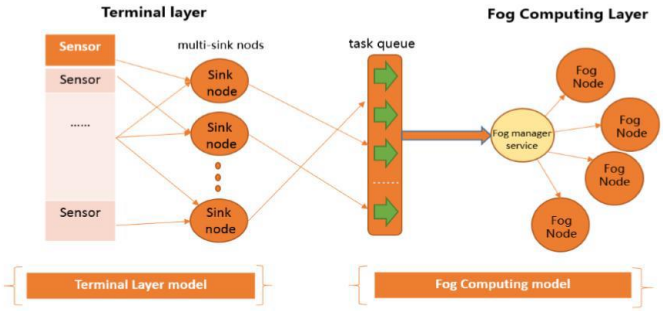

Figure 4. The architecture of the Fog computing model for IoT based model for IoT devices devices.

Table 1. Compact description of the existing work.

Author name	Proposed model	Disadvantages of the model
Yang et al. [28]	They proposed a model that considers circuit, computation, offloading energy consumption to evaluate the overall energy efficiency (EE) in homogeneous fog networks.	The work only focused on the overall energy and did not consider the energy conversions across both the fog and terminal layers.
Pang et al. [29]	They proposed a latency-driven cooperative task computing in multi-user fog–radio access networks, which characterizes the tradeoff between communication and computing across multiple F-RAN (Fog radio access network) nodes.	They did take consider energy consumption problem in both fog layer and terminal layer
Intharawijitr et al. [30]	In terms of the communication distance, they defined a mathematical model of a fog network and the important related parameters to clarify the computing delay and communication delay in fog architecture.	The work did not take into account the energy consumption of the whole model.
Ogawa et al. [31]	The authors presented a use case considering energy consumption measurements of RPL and CTP, and proposed metrics for several scenarios running both RPL and CTP.	The authors did not consider the routing protocol's robustness and reliability.
Felici–Castell et al. [32]	The work focused on analysing different strategies to gather information from different topics. The trade-offs between the "always send" and "local buffer" methods are verified experimentally, which considering power consumption, lifetime, efficiency and reliability.	The reliability of the sink node(s) was not considered.
Machado et al. [33]	The authors proposed a routing protocol based on routing energy and link quality (REL). The end-to-end link quality estimation mechanism, residual energy and hop count are used to select routes to improve the reliability and energy efficiency of IoT applications. In addition, REL proposes an event-driven mechanism to provide load balancing and to avoid premature depletion of energy by nodes/networks.	Their work did not take into account the effect of different number of sink nodes.

Design of FOG Computing Layer and Architecture model for IoT based devices. This model is helpful for transfer of data and to assign resources which are present in the fog layer. This model is named as Fog computing model for IoT based devices (FCM for IoT). As discussed in section III, Figure 3 shows that Fog layer is in between Terminal layer and Cloud layer.

Fog computing model for IoT based devices: Figure 4 shows the proposed architecture of the Fog computing model for IoT based devices (FCM for IoT). The sink nodes of the terminal layer generate the tasks on behalf of the terminal layer nodes and these tasks are kept in the task queue. These tasks are forwarded to the fog manager service routine that is present in the fog layer. This manager service routine assigns the tasks to the fog nodes which will compute the task using one of the existing algorithms.

Proposed Optimal Technique for IoT based devices in FOG Computing Environment. The proposed fog computing layer model basically concentrates on three criterions which reveal the actual functioning of the model. These three criterions are: latency, length and strength. Latency is the time that sink nodes wait after forwarding their request to wait queue. Length is the distance between sink node/user and the corresponding Fog Node and strength is the total strength required by a fog node FN in order to execute the assigned tasks. Let us consider a fog computing layer with n fog nodes labelled as $fn_1, fn_2, \ldots, fn_n$. The tasks that are going to be scheduled on these nodes are $t_1, t_2, \ldots, t_n$. Latency also affects the turnaround time of the task t_k. Where $1 \le k \le n$. Turnaround time is the sum of wait time and execution time. The total length from a sink node to a fog node is calculated by using the formula $TL = \sum_{i=1}^{n} \sqrt{(T_{ix} - fn_{jx})^2 + (T_{iy} - fn_{jy})^2}$. Where $(T_{iy} - T_{iy})$ and (FN_{jx}, FN_{jy}), denote the coordinates of user/sink node T_i and fog node FN_j. Since we are using a fog computing layer to minimize the latency the expected turnaround time is also minimal. There can be more than one task assigned to a fog node. However the turnaround time of a task t by the FN is calculated by using the formula $TAT_t = \max \sum_{i=1}^{n} WT(t_{ij}) + EX(t_{ij})$. Where

TAT_t is the turnaround time of task t, $WT(t_{ij})$ is the wait time of task t_i running on fog node fn_j and $EXE(t_{ij})$ is the execution time of t_i on the fog node fn_j. Saving energy is an important factor which needs to be examined in order to construct a model based on fog computing. Hence the system that is built based on fog computing must have low consumption of energy. The consumption of energy *EXEE* should not be more than electric supply cut-off point. E is the energy consumption for executing task set T by set *FN*. $EXEE = \sum_{i=1}^{n} \sum_{j=1}^{N} EXEE_{ij}$, $EXEE(T, FN) < EL$. Here $EXEE_{ij}$ is the consumed energy of FN_j, $j \in N$ to execute task T_i, $i \in n$ and *EXEE* is the consumed energy of all the *FNs* while executing the assigned tasks; *EL* is the cut-off of energy consumption. In this way depending on the length calculated and energy available the fog manager assigns the computation to the nearest fog node available. This makes the wait time gets decreased and also as a result the Turnaround time also gets decreased.

Table 6.1. Computing power and processing elements in Cloudsim.

Fog Nodes	Node1	Node2	Node3	Node4	Node5	Node6	Node7	Node8
Pes	2	4	2	4	2	2	2	2
Mips	550	300	650	350	750	800	850	900
Energy Cost	10	12	14	15	16	18	20	22
Coordinates	{10, 10}	{10, 40}	{10, 70}	{40, 10}	{70, 10}	{70, 40}	{70, 70}	{60, 80}

Table 6.2. Task Coordinates.

Task No.	Coordinates	Task No.	Coordinates	Task No.	Coordinates	Task No.	Coordinates	Task No.	Coordinates
0	{93,31}	20	{36,75}	40	{73,06}	60	{53,49}	80	{61,21}
1	{32,96}	21	{44,23}	41	{77,45}	61	{52,12}	81	{62,96}
2	{14,11}	22	{31,23}	42	{77,89}	62	{54,11}	82	{64,11}
3	{52,21}	23	{35,23}	43	{72,34}	63	{12,63}	83	{62,48}
4	{50,21}	24	{36,21}	44	{70,21}	66	{10,21}	84	{63,90}
5	{43,90}	25	{33,90}	45	{73,90}	65	{53,93}	85	{67,53}
6	{10,61}	26	{31,21}	46	{77,62}	66	{58,34}	86	{84,70}
7	{96,59}	27	{36,59}	47	{76,59}	67	{50,61}	87	{64,10}
8	{39,83}	28	{49,83}	48	{76,78}	68	{51,24}	88	{63,46}
9	{71,34}	29	{34,51}	49	{11,34}	69	{42,83}	89	{12,37}
10	{23,31}	30	{43,31}	50	{83,31}	70	{43,51}	90	{13,14}
11	{22,96}	31	{32,96}	51	{92,96}	71	{44,67}	91	{39,57}
12	{24,11}	32	{14,11}	52	{96,75}	72	{44,11}	92	{17,11}
13	{23,83}	33	{59,39}	53	{92,21}	73	{49,87}	93	{52,31}
14	{20,21}	34	{54,52}	54	{95,24}	74	{44,59}	94	{50,61}
15	{23,90}	35	{43,90}	55	{93,90}	75	{53,12}	95	{44,23}
16	{28,95}	36	{10,61}	56	{90,61}	76	{40,61}	96	{13,71}
17	{26,59}	37	{96,59}	57	{45,32}	77	{49,56}	97	{95,69}
18	{66,66}	38	{57,74}	58	{99,83}	78	{49,83}	98	{32,53}
19	{28,45}	39	{75,23}	59	{68,21}	79	{41,34}	99	{63,31}

Experimental Results

Simulation work is carried out on Cloudsim for the proposed Fog computing model. Cloudsim is based on the existing architecture of Gridsim which can run on windows and Linux. The following results are carried out on a intel Pentium Dual Core P6000 processor with 8GB RAM and 1TB HDD and the OS used is Windows 7. There are 8 fog nodes in the Cloudsim environment. Following is the table 6.1 that shows computing power and processing elements at each node in Cloudsim. The coordinates of FNs are simulated in an area, such as a building or a city, hence we limited the range of FN in 0-100, and coordinates of these nodes are shown in the following table 6.2.

Latency in Processing

Figure 6.1 shows the latency caused by three algorithms. These results show a considerable reduction in latency while processing thus it shows increased performance in computation and energy consumption. However if we consider more than 100 tasks also the proposed work shows good performance as the three lines in the graph are meeting after 100 tasks.

Length to Fog Nodes

Figure 6.2 shows the lengths from users to corresponding fog nodes. From the figure it is evident that FCM for IoT gives better results when compared to Fog Oriented MaxMin and MaxMin algorithms. The gentleness of the line indicates that there is not much change in the length as the number of tasks increasing and the length can be predicted in FCM for IoT.

Consumption of Energy

The factor that is given considerable importance is Consumption of Energy. In this work the energy consumption required by all the nodes is taken into account and the same is shown in Figure 6.3. The consumption of energy is measured in milliamp ere hour and from the figure this energy consumption is less in FCM for IoT when compared to Fog Oriented MaxMin and MaxMin algorithms.

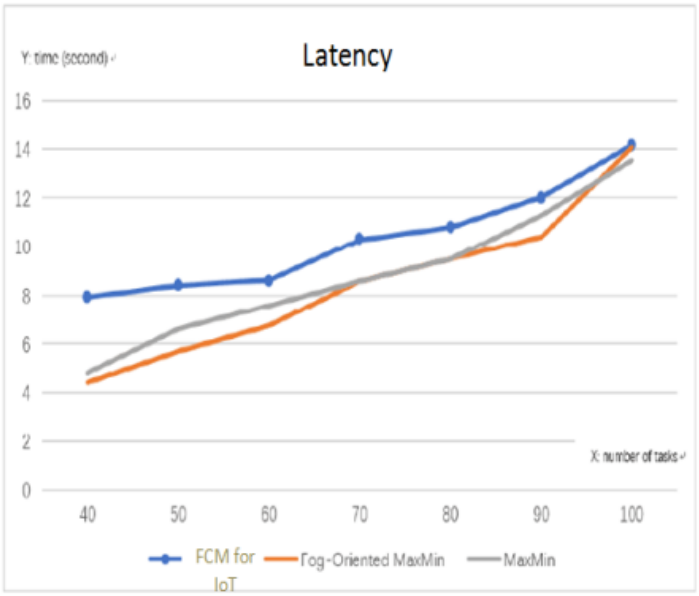

Figure 6.1. Latency.

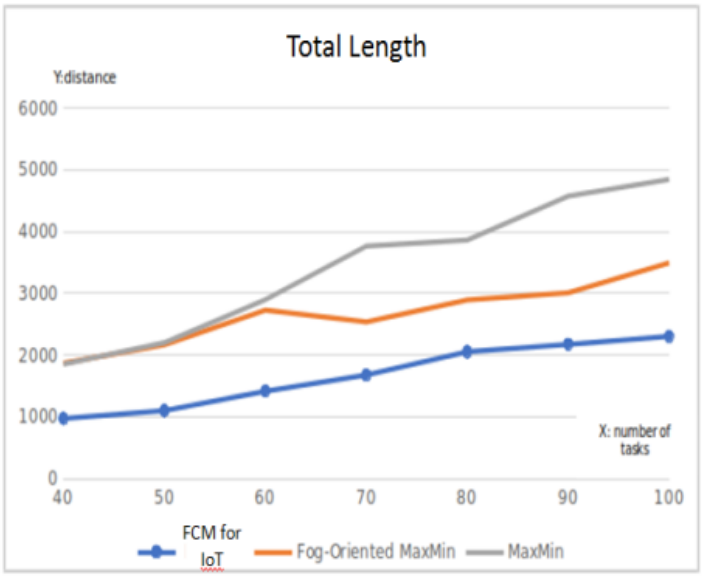

Figure 6.2. Total length.

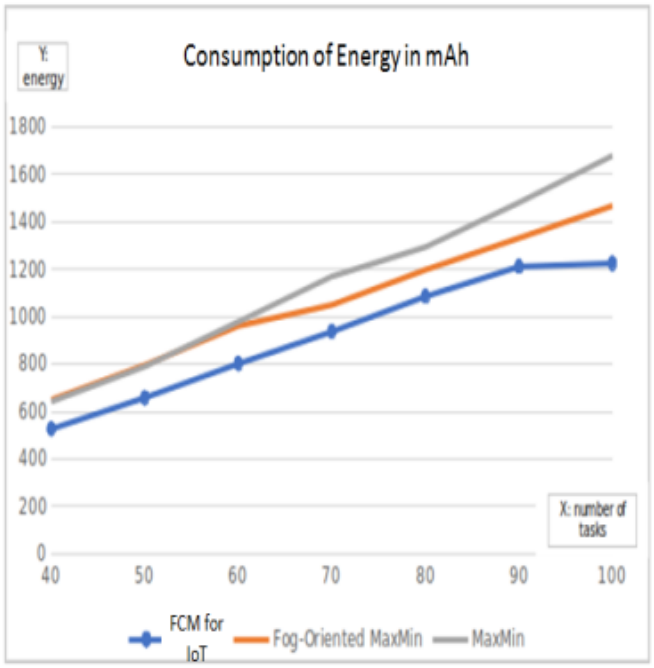

Figure 6.3. Total consumption of energy.

Conclusions and Future Work

In this work I have identified few major lineaments that are given considerable importance in order to reduce the kind of threats that are taking place in IoT World. The proposed model FCM for IoT helps in reducing Latency time, Turnaround time of the tasks by including a Fog Layer in between Cloud Computing Layer and Terminal layer. The results that are obtained demonstrate that few of the lineaments like Correlation, Captivity, Inattentiveness and Consumption of Energy are reduced in the proposed FCM for IoT.

The other major lineaments may cause different kinds of attacks hence further research work is to be carried out for reducing the attacks that will take place because of the presence of variety, unaccountability, mobility and Omni presence.

References

[1] The Statistics Portal, 80 IOT Statistics (Infographic) Dreams of a connected world 80 Internet of Things Stats and Facts [Online} Available: https://safeatlast.co/blog/iot-statistics/

[2] Lin, Jie, et al., A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications, IEEE Internet of Things Journal, vol. 99, 2017.

[3] Li, Shancang, T. Tryfonas and H. Li, The Internet of Things: a security point of view, Internet Research 26(2) (2016), 337-359.

[4] S. Sicari, et al., Security, privacy and trust in Internet of Things: The road ahead, Computer Networks the International Journal of Computer and Telecommunications Networking 76 (2015), 146-164.

[5] Yang Yuchen, et al., A Survey on Security and Privacy Issues in Internet of Things, IEEE Internet of Things Journal 4(5) (2017), 1250-1258.

[6] W. Trappe, R. Howare and R. S. Moore, Low-energy security: Limits and opportunities in the Internet of Things, IEEE Security Privacy, vol.13,no 1, pp, 14-21, Jan/Feb, 2015.

[7] Fu, Kevin, et al. (2017). Safety, Security, and Privacy Threats posed by accelerating trends in the Internet of Things. Technical Report. Computing Community Consortium. [online] Available:http://cra.org/ccc/wp-content/uploads/sites/2/2017/02/Safety-Security-and-Privacy-Threats-in-IoT.pdf.

[8] R. Roman, J. Zhou and J. Lopez, On the features and challenges of security and privacy in distributed Internet of Things, Comput. Netw. 57(10) (2013), 2266-2279.

[9] The University of Texas Portal, The problem of Over privilege in IoT Platforms (Online} Available: http://www.ece.utexas.edu/events/problem-over-privilege-iot-platforms

[10] Jia and Yunhan Jack, et al., Contex IoT: Towards providing contextual integrity to applied IoT platforms, Network and Distributed system security symposium 2017, pp. 1-15.

[11] Liu and Hui, et al., Smart solution poor protection: An empirical study of devices, IoT Security and Privacy workshop (2017), 13-18.

[12] Chen and D. Daming, et al., Towards automated dynamic analysis for Linux based embedded firmware, Network and distributed system security symposium, 2016.

[13] Zhao and Lu, et al., ARMor: fully verified software fault isolation, Proceedings of the International Conference on Embedded Software IEEE, 2011:289-298.

[14] Schulz, Patrick Koeberl Steffen, Ahmad-Reza Sadeghi and Vijay Varadharajan, Trustlite: A security architecture for tiny embedded devices, EuroSys. ACM, 2014, 1-14.

[15] Clements and A. Abraham, et al., Protecting Bare-Metal Embedded Systems with Privilege Overlays, Security and Privacy IEEE, 2017.

[16] Chung and Taegyu et al., Securing Real-Time Microcontroller Systems through Customized Memory View Switching, Network and Distributed System Security Symposium, 2018.

[17] Checkpoint Research, IoTroop Botnet: The Full Investigation, [Online]. Available: https://research.checkpoint.com/iotroop-botnet-full-investigation/ (2017).

[18] M. P. Paet al, IoTPOT: Analysing the rise of IoT compromises, Proc. USENIX Conf. Offensive Technol. (2015), 1-9.

[19] C. Zhang and R. Green, Communication security in Internet of Thing: Preventive measure and avoid DDoS attack over IoT network, Proc. Symp. Commun. Netw. Soc. Comput. Simulat. Int. (2015), 8-15.

[20] P. Kasinathan, G. Costamagna, H. Khaleel, C. Pastrone and M. A. Spirito, DEMO: An IDS framework for Internet of Things empowered by 6LoWPAN, in Proc. ACM Sigsac Conf. Comput. Commun. Security (2013), 1337-1340.

[21] Z. Lu, W. Wang and C. Wang, Camouflage traffic: Minimizing message delay for smart grid applications under jamming, IEEE Trans. Dependable Secure Comput. 12(1) (2015), 31-44.

[22] Guan and Le et al., Trust Shadow: Secure execution of unmodified applications with ARM trustzone, Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services. ACM, 2017.

[23] Copos and Bogdan, et al., Is Anybody Home? Inferring Activity From Smart Home Network Traffic, Security and Privacy Workshops IEEE, (2016), 245-251.

[24] Yang and Weining, et al., Minimizing private data disclosures in the smart grid, ACM Conference on Computer and Communications Security. ACM, (2012), 415-427.

[25] Li and F. Fengjun, Li and F. Li, A multi-cloud based privacy-preserving data publishing scheme for the internet of things, Conference on Computer Security Applications ACM, (2016), 30-39.

[26] Chen, Ing Ray, F. Bao and J. Guo, Trust-based Service Management for Social Internet of Things Systems, IEEE Computer Society Press, 2016.

[27] Wright and Alex, Mapping the internet of things, Communications of the ACM, 60(1) (2016), 16-18.