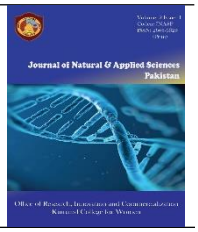
Contents lists available http://www.kinnaird.edu.pk/

Journal of Natural and Applied Sciences Pakistan

Journal homepage: http://jnasp.kinnaird.edu.pk/



USING FOG COMPUTING FOR SECURITY AND STORAGE CHALLENGES IN IOT

Nageen Saleem[1], Dr. Muhammad Rizwan[1*], Dr. Fahad Ahmad[1]
[1]Department of Computer Science, Kinnaird College for Women, Lahore.

Article Info

*Corresponding Author
Tel: +92 333-4881501
Email:
muhammad.rizwan@kinnaird.edu.pk

Keywords
Fog Computing, Internet of thing (IoT), Cloud Computing, Wireless Networks, Real Time Systems, Certificate Revocation Lists (CRLs), Certificate Revocation Schemes (CRS).

Abstract

Fog computing broadens the concept of cloud computing realm providing enhanced services in terms of storage and network, giving growth to use of new versions of applications and services. This paper covers the characteristics of fog computing. It elaborates either the use of fog computing is apt for number of critical internet of things. (IoT) services and applications namely Smart Cities i.e. Wireless Sensor Networks and how these characteristics have increase the security and reliability issues for end users. It covers how fog computing has enhanced its use in IoT and the recent security concerns regarding its use. Lastly, it defines the possible solutions to overcome the challenges of reliability and enhance the use of fog computing. A proposed solution is the security issue to be handled by a Third Party completely and improved certificate cancellation procedure among IoT devices for security enhancement.

1. Introduction

Cloud Computing is the delivery of computing services over the Internet referred as Cloud, services include storage, networking etc. As the cloud computing has become to be a source of high productivity, performance, increased speed and cost effective service providence, It has created some problems delay-sensitive applications which requires nodes to meet the delay requirements. As the modern era has stepped forward with Internet of Things (network of physical devices embedded with electronics, software and sensors), it requires enhanced support, location awareness, decreased delay and scalable networks answering geo distribution problems.

This research represents a latest platform to answer the problems unhandled by cloud computing is Fog Computing also known as fogging, introduced by Cisco, a technique used to bridge the gap between remote data centers and IoT devices. It extends the cloud to the

network edge as fog which literally means something which is closer than a cloud and provide efficient data access, computation and storage facilities. Fog Computing advances in the paradigm of cloud (Alrawais *et al*, 2017) with respect to location awareness, geo distribution, large scalability, low latency and heterogeneity. This paper explains how fog and cloud intermingle to provide better data management and analytical services. The basic purpose of fog computing is to enhance Quality of Service (QoS) and decrease traffic on cloud servers (Bonomi *et al*, 2012). As the use of IoT devices have increased, these come up with challenges of decreased latency and computation power, battery and storage as well, which overall affect the user experience. This research claims that how Fog Computing is able to address all these problems with its unique characteristics such as territorial awareness and area scalability. Several works (Hwang *et al*, 2009) have been done on applying Fog Computing such as health care systems, vehicles and ad hoc networks.

As with emergence of these IoT devices, authentication has played a vital role in in establishing the relations between fog nodes and these devices. Trust in this case plays a vital role in developing this kind of relationship. The two-way challenge in FogNet where the nodes need to maintain trust relationship with IoT devices is a formidable concern. Similarly, Fog Computing could be easily attacked by Man-In-The-Middle and unfold sensitive information of sensor nodes (Hwang *et al*, 2009). Fog computing authentication and detection system issues need to be exploited as well. This research unveils the security challenges faced by Fog Computing and how a Third Party could be a better solution in addressing these issues.

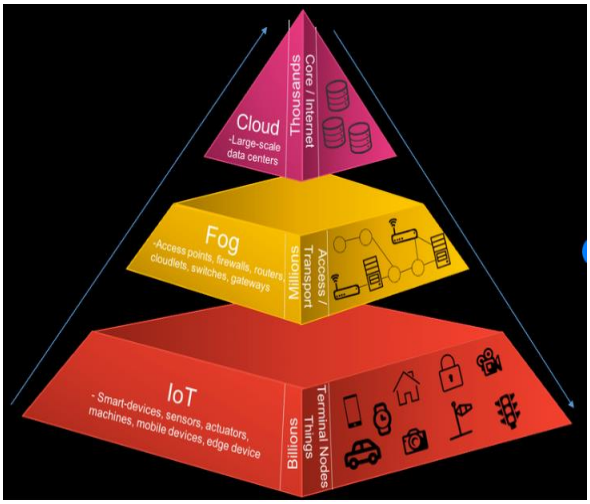


Figure 1: Cloud based CRVANET Challenges

1.1.Architecture and Characteristics of Fog Computing

As fog computing extends the conventional idea of cloud computing and services to the edge node of the network. It provides better facilities from computation to storage and communication management, reducing the risk factor as well. The hierarchical architecture (Prakasha *et al*, 2018) as shown in Figure 1 comprises of three layers:

- First being the *Terminal layer* which consists of IoT devices. Each device is territorially distributed. This layer is nearest to the physical medium. It takes the data from physical medium and then transfers the information to the upper layer.
- The next layer is *Fog Layer* comprising of fog nodes (routers, gateways) and being

located at the edge of the network. It is these nodes which are colossally distributed between the cloud and end devices (shopping centers, streets, departmental stores). These end devices communicate with fog nodes to avail maximum services. Fog nodes provide latency free services and are connected to cloud data center via IP networks.

- The third layer is *Cloud Computing Layer* which has the tendency of extensive computation and permanent storage as it consists of multiple servers and storage devices.

These layers provide all the services demanded in IoT devices i,e high bandwidth and low latency and increased geographical distribution. However, there remain some issues still unhandled in all the layers. Here is where the security and integrity issues arise and goes unhandled. Terminal layers as it consists of all the IoT devices interlinked and intertwined. It is called terminals as these are in the real time environment fetching the data and transferring it to next layers. If a device due to certain hazardous conditions not work appropriately it means that it is damaged at the terminal end and the issue needs to be handled at the terminal layer. The fog layer is in constant contact with the terminal layer and hence is responsible for the establishment of connection and exchange of information amongst layers. It is connected at the edge of network. The cloud layer maintains, handles and responds to the change if any in the network and connectivity phase. Being the handler, there then appears certain security gaps difficult to fill by these layers alone in IoT.

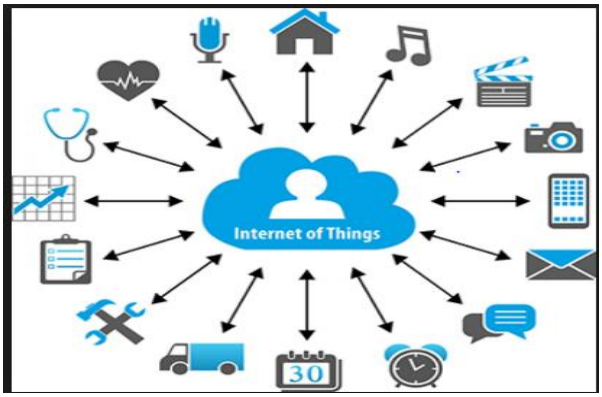


Figure 2: Cloud Computational Hierarchical Architecture In CRVANET

1.2.Security and Private Challenges in Iot

Security IoT consists of networked things (Byron *et al*, 2018) generally wireless sensors and actuators each connected with gateways and as the data is digitalized and lastly comes across cloud and edge processing. Internet of Things (IoT) refers to the stringent connectedness between digital and physical world as shown in Figure 2. IoT advances in the field of service provision and usage of digital technology. However, there are still some security issues unaddressed as IoT lacks in memory and CPU power to perform the cryptographic operations (Ma *et al*, 2006) needed for authentication. The level and degree of Trust of these IoT devices is also to be taken into account of. Several works (Corcoran *et al*, 2016) have been done to improve the trust level of cloud building models. One of the characteristics of IoT is Rouge Node (Bonomi *et al*, 2012) which is capable of deteriorating users' data.

The privacy problem of the IoT devices (Fan *et al*, 2014) is also vulnerable threat to growing use of IoT in the global world. Moreover, the handling, protection and access of data is also another significant issue to be dealt with. Fog Computing can come up with a better solution to the reliability and security issues being faced by the IoT devices such as DoS (Denial of Service attacks) and malware based attacks. There are various schemes to address the issue. Most recent being certificate revocation schemes. However, this CRSs have limited efficient mechanism which include increased bandwidth and time space constraints. Some basic issues are listed in Table 1 in detail.

Table 1: Challenges faced by IoT devices

Smart Home	• Limited Authentication, authorization, and accounting services provided by the companies. • Security issues in web based interfaces • Lacks in cryptographic support
Smart City	• Restricted privacy • Integrity issues • Connectivity issues • Lacks in cryptographic support
Smart Health	• Restricted privacy • Integrity issues • Connectivity issues • Lacks in cryptographic support • Limited Availability

2. Literature Review

There are some deep and significant researches carried on the significant issues of Fog Computing and Security. Alrawais *et al*, 2017 highlights the security issues being faced by Fog Computing and how the use of bloom filter can reduce the cost and latency of the system. It's a detailed overview of the security issues being faced by IoT devices and how they are using the Certificate Authority for generation of a small list using the data structure of broom filter. However, using bloom filter is practically less efficient. Bonomi *et al*, 2012 elucidates the increased role of Fog computing in IoT. Prakasha *et al*, 2018 and Byron *et al*, 2018 elucidates in detail the fog architecture and Internet of things respectively. Cloud Security and building of trust (Hwang *et al*, 2009) is also a significant study carried. One of the conference proceedings, Usenix tells how PKI has played a role in CRLs creation as Public Key infrastructure which provides a set of roles, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates and manage public-key encryption. Gubbi et *al*, 2013 also in detail examines about the certificate revocation lists and their use in the digital world. Fan proves with evidence how practically cuckoos filter is better than bloom filter which means as both are better data structures However, practically cuckoo's filter is a better approach due to its quality of using hashing in buckets and reducing storage and simultaneously checking the value being placed at correct position. Corcoran *et al*, 2016 provides a detailed framework of privacy of Fog computing where it has been declared as one of the emerging technologies being used in IoT devices. It also highlights various aspects of recent researches which has played an important role in telling the trend of Fog computing and how it is taking over cloud computing architecture due to its better and probabilistic

future security patterns. Stojmenovic *et al*. also highlights the working architecture and recent security scenarios in Fog Computing It depicts the role of Fog in the ubiquitous computing and how the recent development may lead to increased security and privacy issues. Hu *et al*, 2017 presents the process of hash tables insertion being improved. It describes how these techniques can be enhances in performance.

3. Problem Statement

As the fifth generation of Internet and networks come into existence, the idea of IoT seems to be ubiquitous and near to implementation even in the third world countries. The relationship between Fog and the network nodes i.e the wireless sensors used in smart cities holds vital importance as Fog computing reduces the problems of delay and scalable distribution.

Although Fog computing with in F-RANs i.e integrating Fog with RAN and creation of Certificate revocation lists (Gubbi et *al*, 2013) provide better security optimization properties, However, the problem of complete authenticity and limitations of Certificate revocation lists as they demand a huge space from the end user increasing overhead.

There are various methods predefined for creation of CRLs. However this creation need an increased and huge space from clients side and there is still much chances of improvement in creation , authentication and space management once the CRL is being created using Fog and Cloud. We would answer the improved version of issues of space and security during creation of CRL and the recent researches based on the issue. We would cover how to enhance the security factor of the technique as an authentic exchange of messages and protecting privacy of end user is the demand acquired by all the IoT devices in linked with Fog Computing.

The need to develop more security concerned clouds and fogs has been increasing and various ways are being determined to provide effective methods to ensure privacy, authenticity and efficiency of these techniques which serve to be a better storage and other service alternatives.

4. Overview of the Proposed Scheme

Our proposed solution is depicted in Figure 3 in flow chart that a Third Party working at the back end of cloud, fog and IoT will create digitalized certificates creation using one of the Public Key Infrastructure Characteristics. One fog node can be responsible for various certificates and vice versa. In the proposed solution, we use Cuckoos filter which is a reliable space and probability efficient data structure unlike bloom filter (Sklavos *et al*, 2016) used in (Hwang *et al*, 2009). We use a cuckoos filter which is a space and time efficient data structure better than bloom filter (Hwang *et al*, 2009) to create list that can decrease the revocation list data size and overhead and then it can be used to check whether an element is a member of the group. Figure 3 shows the flow of our proposed scheme.

Initially, cuckoos filter vector is empty and to keep the record of CR, we use their unique serial numbers assigned by a Trusted Third Party.

Cuckoos filter with usage of hash tables shown in Figure 4 will store all the CRLs information of serial number and lower down the space, efficiency cost overheads of the IoT using Fog and Cloud. Trusted Third Party will be responsible for creation, cancellation and updating of any certificate.

Moreover, the large data to be held on client side will be turned over to Fog and there will be a latency free communication between Third Party, Cloud, Fog nodes and IoT devices. The cost of space is reduced as Cuckoos filter is better than bloom filter. With this the security issue will be handled by a separate entity which would be wholly responsible for security maintenance of the architecture. The Trusted Party uses PKI for creation and signing of CRLs and hence sends the created lists to the cloud and the cloud further gives the received information to the fog nodes as each fog node deal with different IoT. The Fog then stores the list and prepares a Cuckoos filter that maps the data of revocation received. The Fog then is responsible for the assigning of filter to the specific IoT device.

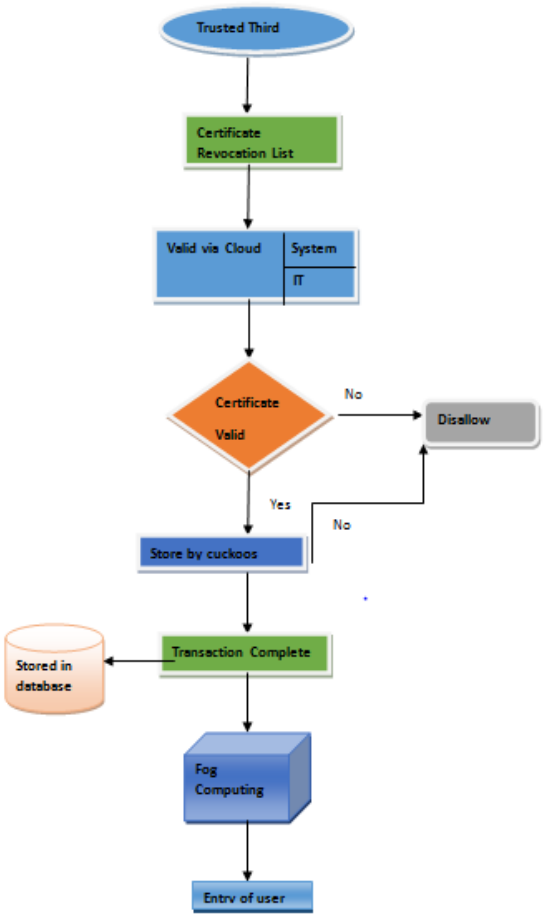


Figure 3: Flow chart of proposed scheme

Each IoT then authenticate the fog's signature and store the given CRLs. During communication with other device, IoT needs to validate the device's certificate status. If there exist no certificate in the filter it means it is not cancelled which means another will be created using insertion function of the cuckoos filter. But once the certificate is found the filter gives a false positive response to the Fog which then allow all IoT to verify the certificate. IoT devices send information of certificate serial number against the stored information and replies with status.

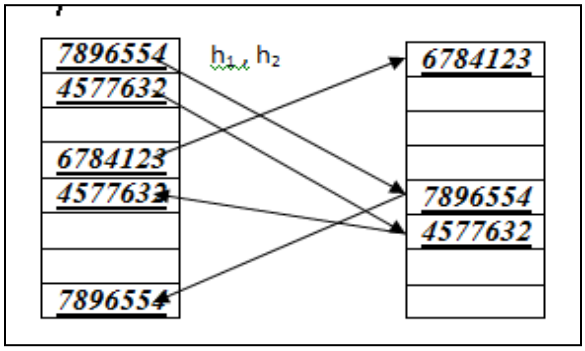


Figure 4: Hasing in cuckoos filters

4.1.Proposed Algorithm

The following algorithm shows how the cuckoos filter hashing work. Each serial number is checked and then replaced in the two hashes using buckets. It is checked that if one bucket is filled then replace the value in the other hash function using hash tables (Hu *et al*, 2017) and vice versa. It keeps filling the value until the buckets are fill and then a full short list of member groups is prepared and sent to the fog nodes.

Parameters for the filter:

Algorithm for the hashing function:

Let the functions be h1 and h2

An array with n buckets is taken

// The i-th bucket will be called A[i]

// Inputs are taken Let CRLs be the list of elements be inserted into the filter.

While K is not empty

Enter the value from List CRLs into the array

If A [h1] is empty place the S2 in A [h1]

Else if A [h2] is empty place the S2 in A[h2]

Else Let S2 be another element in A [h2]

Prepend S2 to CRLs

And place S1 into A2 [h2]

5. Performance analysis and discussion

The idea is to develop certain IoT devices which must be less challenging in terms of security issues and storage problems. With the advent of computing, there are two big most sensitive issues. The security issue in the proposed solution is handled by a Trusted Third Party which creates Certificate Revocation Lists via cloud for the member devices which are trust worthy and which are prone to malicious attacks. It then with the help of a special data structure creates a short list which would reduce the storage cost of the devices. The proposed architecture is resource and security efficient. There is delay sensitive, less vulnerable to attacks and gives the storage edge as well compared to various other works being done recently over the issue. There is instant updating of cancellation of certificate information and creation if new information. Moreover, the Third Party will be solely responsible for the constant communication, transfer of messages, exchange of information, handling of data and error free fetching of information using the characteristics of Public Key infrastructure for CRLs creation (Stojmenovic *et al*, 2014). With creation of CRLs, it would also make sure the architecture being protected and not attacked by any sudden or unexpected attack. There are some detailed works on cuckoos filter as false probability though less than bloom filter is something still needs to be worked on which makes cuckoos

filter practically better approach to be used and relied upon. However, when its about the space complexity cuckoos filter is of less space complexity than bloom as

$$\log\left(\frac{1}{fpp}\right) + 2 \Big/ load$$ load = bits per entry

fpp= false probability value

As the bloom filter's false positive probability rate is always increasing, the cuckoos filter provide a limited rate which makes it more suitable for the functions like holding data of serial numbers of certificate revocation lists. Hashing between the two tables using hash functions will make sure the appropriate CRLs created is in its specified node and right space is utilized. The security problem is also dealt by the Third Party which will be solely responsible for any reliability and privacy issues, hence shifting the whole cost to one domain either than distributing it over certain other aspects.

High-speed approximate set-membership tests are critical for many applications, and filters are commonly used for the specific purpose, however bloom filter does not support deletion but for approximate set-membership test applications the most suitable is cuckoos filter. This cuckoo filter allows dynamic removal and addition of values achieving high performance as shown in the performance analysis.

Cuckoos filter provide much lower false positive rates with the storage issue widely reduced comparatively with bloom filter. Figure 5 and 6 show the analysis referred from fast forward labs comparing the two probabilistic data structures.

5.1.Space and Complexity

Relating to both filters, they perform differently at different false positive probabilities. As the false positive probabilistic value is less than or equal to 3 percent, the cuckoos filter has less bits per entry. However, at a higher rate it blooms filter has fewer bits per entry.

5.2.Time Complexity

In cuckoos hashing, insertion of an element seems to be more than O (1) times because there is more rate of collision, to remove a value and make another room for the value. In these cases the whole table may be reversed and rehashed as well.

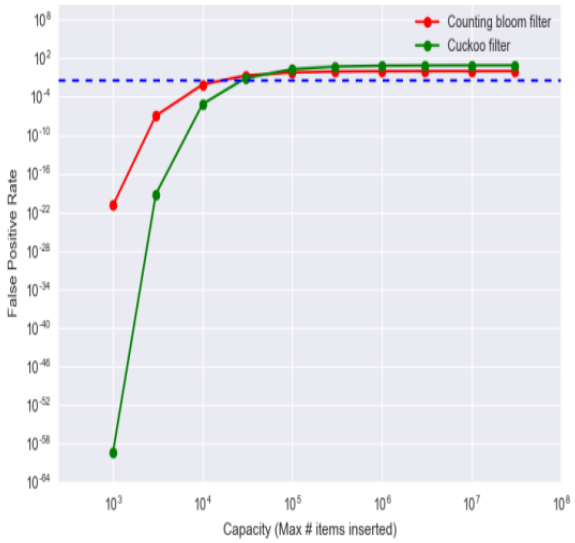


Figure 5: Capacity of insertion of both filters

The speed of cuckoos filter is three times faster than bloom filter while inserting the items, it was observed. There's a significant increase in throughput for insertion in cuckoos filter. Although by the optimization of Bloom filter, the insertion throughput can be increased.

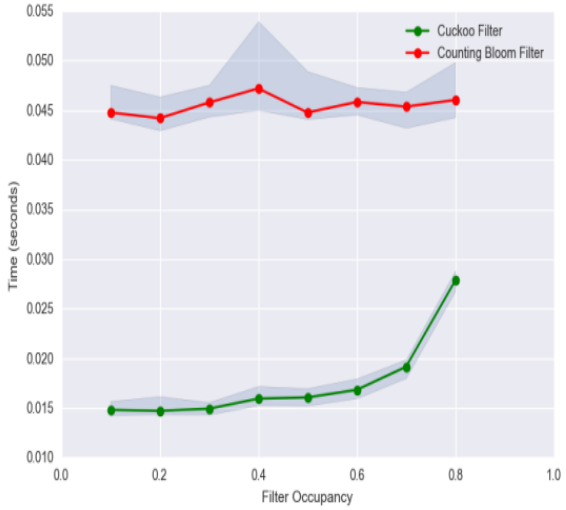


Figure 6: Cuckoos filter throughput

6. Conclusion and Discussion

This research unfolds the Fog Computing architecture, its emerging role in IoT, the possible and present security problems faced by the end user and a way forward i.e. Third Party involvement which would improve the security and space complexity to cater the problem of protection of privacy and integrity. The purpose of this research is to provide a detailed view of Fog computing being used in IoT devices and how to decrease the security and reliability challenges so that future generations can benefit from the solution and the end user must be satisfied with security and latency issues being catered to be diminished as much as possible.

A proposed solution is to assign the Third Party the role to be responsible for CRLs using an efficient data structure cuckoos filter and reduce the security overhead by providing better security environment for the architecture to work and communicate. The motivation behind is the recent researchers in the field of Fog security and how the methodology needs improvement.

References

Alrawais, A., Alhothaily, A., Hu, C., & Cheng, X. (2017). Fog Computing for Internet of Things: Security and Privacy issues. *IEEE Computer Society*, 21(2), 34-42.

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog Computing and Its role in Internet of Things. *Cisco*, 1-3.

Byron, M., Christian, M., Sameh, G., Ren, W., & Tsung-yuan CT, (2018). Hash table entries insertion method and apparatus using virtual buckets. United States Intel Corporation (Santa Clara, CA, US) 20180109460, http://www.freepatentsonline.com/y2018/0109460.html.

Corcoran, P. M. (2016). A privacy framework for the Internet of Things. *IEEE 3rd World Forum on Internet of Things (WF-IoT)*, Reston, VA, 13-18, doi: 10.1109/WF-IoT.2016.7845505.

Fan, B., Andersen, D.G., Kaminsky, M., & Mitzenmacher, M.D. (2014). Cuckoo Filter: Practically Better Than Bloom. , *CoNEXT '14: Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, 75–88. https://doi.org/10.1145/2674005.2674994.

Gubbi, J., Buyya, R., Marusic,S., & Palaniswami, M. (2013). Internet of

Things (IoT): A vision, architectural elements, and future directions. Elsevier, 29(7), 1645-1660.

Hu, P., Dhelim, S., Ning, H., & Qiu, T. (2017). Survey on fog computing: architecture, key technologies, applications and open issues. *Journal of Network and Computer Applications*, 98, 27-42.

Hwang, K., Kulkareni, S., & Hu,Y. (2009). Cloud Security with Virtualized Defense and Reputation Based Trust Management. *Proceedings of the 2009 Eighth IEEE International Conference on Dependable, Autonomic and Secure Computing*, 717–722. https://doi.org/10.1109/DASC.2009.149

Ma, C., Hu, N., & Li, Y. (2006). On the Release of CRLs in Public Key Infrastructure. *15th Usenix Security Symposium*, 2.

Prakasha, K., Muniyal, B., Acharya, V., Krishna, S., & Prakash, S. (2018). Efficient digital certificate verification in wireless public key infrastructure using enhanced certificate revocation list. *Information Security Journal*, *27*(4), 214-229. https://doi.org/10.1080/19393555.2018.1516836.

Sklavos, N., & Zaharakis, I.D. (2016). Cryptography and Security in Internet of Things (IoTs): Models, Schemes, and Implementations. *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, 1-2.

Stojmenovic, I., & Wen, S. (2014). The Fog Computing Paradigm: Scenarios and Security Issues. *Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, ACSIS* , 2, 1-8.