CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2016; **28**:2991–3005
Published online 29 April 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.3485

SPECIAL ISSUE PAPER

An overview of Fog computing and its security issues

Ivan Stojmenovic, Sheng Wen*,†, Xinyi Huang and Hao Luan

School of Information Technology, Deakin University, VIC 3125, Australia

SUMMARY

Fog computing is a paradigm that extends Cloud computing and services to the edge of the network. Similar to Cloud, Fog provides data, compute, storage and application services to end users. In this article, we elaborate the motivation and advantages of Fog computing and analyse its applications in a series of real scenarios, such as Smart Grid, smart traffic lights in vehicular networks and software defined networks. We discuss the state of the art of Fog computing and similar work under the same umbrella. Distinguished from other reviewing work of Fog computing, this paper further discloses the security and privacy issues according to current Fog computing paradigm. As an example, we study a typical attack, man-in-the-middle attack, for the discussion of system security in Fog computing. We investigate the stealthy features of this attack by examining its CPU and memory consumption on Fog device. In addition, we discuss the authentication and authorization techniques that can be used in Fog computing. An example of authentication techniques is introduced to address the security scenario where the connection between Fog and Cloud is fragile. Copyright © 2015 John Wiley & Sons, Ltd.

Received 8 November 2014; Revised 3 February 2015; Accepted 3 March 2015

KEY WORDS: Fog computing; security

1. INTRODUCTION

Cisco recently delivered the vision of Fog computing to enable applications on billions of connected devices, already connected in the Internet of Things (IoT), to run directly at the network edge [1]. Customers can develop, manage and run software applications on Cisco IOx framework of networked devices, including hardened routers, switches and Internet protocol video cameras. Cisco IOx brings the open source Linux and Cisco IOS network operating system together in a single networked device (initially in routers). The open application environment encourages more developers to bring their own applications and connectivity interfaces at the edge of the network. Regardless of Cisco's practices, we first answer the questions of what the Fog computing is, and what are the differences between Fog and Cloud.

In Fog computing, services can be hosted at end devices such as set-top-boxes or access points. The infrastructure of this new distributed computing allows applications to run as close as possible to sensed actionable and massive data, coming out of people, processes and thing. Such Fog computing concept, actually a Cloud computing close to the 'ground', creates automated response that drives the value.

Both Cloud and Fog provide data, computation, storage and application services to end users. However, Fog can be distinguished from Cloud by its proximity to the end users, the dense geographical distribution and its support for mobility [2]. We adopt a simple three level hierarchy as

*Correspondence to: Sheng Wen, School of Information Technology, Deakin University, VIC 3125, Australia.
†E-mail: wesheng@deakin.edu.au

Copyright © 2015 John Wiley & Sons, Ltd.

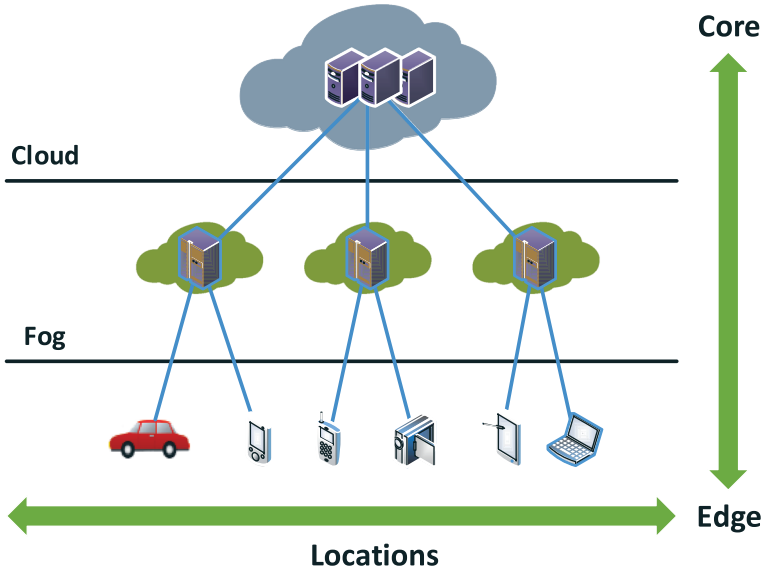

Figure 1. Fog between edge and cloud.

in Figure 1. In this framework, each smart thing is attached to one of the Fog devices. Fog devices could be interconnected, and each of them is linked to the Cloud.

Three overviews of Fog computing paradigm have been published recently [3–5]. They provide comprehensive definition of Fog computing and comparisons to Cloud. To be distinguished, this overview takes a close look at the Fog computing paradigm by real application scenarios. This part of research is to investigate Fog computing advantages for services in several domains, such as Smart Grid, wireless sensor networks, IoT and software defined networks (SDNs). We examine the state of the art and disclose some general issues in Fog computing such as service migration among Fog devices and between Fog and Cloud.

Another unique contribution of this overview is to provide a comprehensive discussion on the security issues in Fog computing. Fog device is itself vulnerable to attacks, such as man-in-the-middle attack [6]. Therefore, we first analyse the system security issues according to current Fog computing paradigm. We provide experiments on the central processing unit (CPU) and memory consumption to investigate the stealthy features of the attacks. We further analyse the security issues among Fog devices and between Fog and Cloud. The authentication and authorization techniques will be discussed in this part of research. This paper is an extension of our previous paper [7] with more security issues.

This paper is structured as follows. Section 2 briefly explains the significance of Fog computing, followed by Section 3, which illustrates various scenarios of Fog computing. Section 4 introduces the state of the art and similar work under the same umbrella. Security issues are elaborated in Sections 5 and 6. Section 7 concludes this article with a discussion of future work.

2. WHY DO WE NEED FOG?

In the past few years, Cloud computing has provided many opportunities for enterprises by offering their customers a range of computing services. Current ‘pay-as-you-go’ Cloud computing model becomes an efficient alternative to owning and managing private data centres for customers facing Web applications and batch processing [8]. Cloud computing frees the enterprises and their end users from the specification of many details, such as storage resources, computation limitation and network communication cost. However, this bliss becomes a problem for latency-sensitive applications, which require nodes in the vicinity to meet their delay requirements [2]. When techniques and devices of IoT are getting more involved in people’s life, current Cloud computing paradigm can hardly satisfy their requirements of mobility support, location awareness and low latency.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

Fog computing is proposed to address the aforementioned problem [1]. As Fog computing is implemented at the edge of the network, it provides low latency, location awareness and improves quality-of-services (QoS) for streaming and real time applications. Typical examples include industrial automation, transportation and networks of sensors and actuators. Moreover, this new infrastructure supports heterogeneity as Fog devices include end-user devices, access points, edge routers and switches. The Fog paradigm is well positioned for real time big data analytics, supports densely distributed data collection points and provides advantages in entertainment, advertising, personal computing and other applications.

3. WHAT CAN WE DO WITH FOG?

We elaborate on the role of Fog computing in the following six motivating scenarios. The advantages of Fog computing satisfy the requirements of applications in these scenarios.

Smart Grid: Energy load balancing applications may run on network edge devices, such as smart metres and micro-grids [9]. Based on energy demand, availability and the lowest price, these devices automatically switch to alternative energies like solar and wind. As shown in Figure 2, Fog collectors at the edge process, the data generated by grid sensors and devices, and issue control commands to the actuators [2]. They also filter the data to be consumed locally and send the rest to the higher tiers for visualization, real-time reports and transactional analytics. Fog supports ephemeral storage at the lowest tier to semi-permanent storage at the highest tier. Global coverage is provided by the Cloud with business intelligence analytics.

Smart Traffic Lights and Connected Vehicles: A video camera that senses an ambulance flashing lights can automatically change street lights to open lanes for the vehicle to pass through traffic. Smart street lights interact locally with sensors and detect presence of pedestrian and bikers, and

Figure 2. Fog computing in smart grid.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

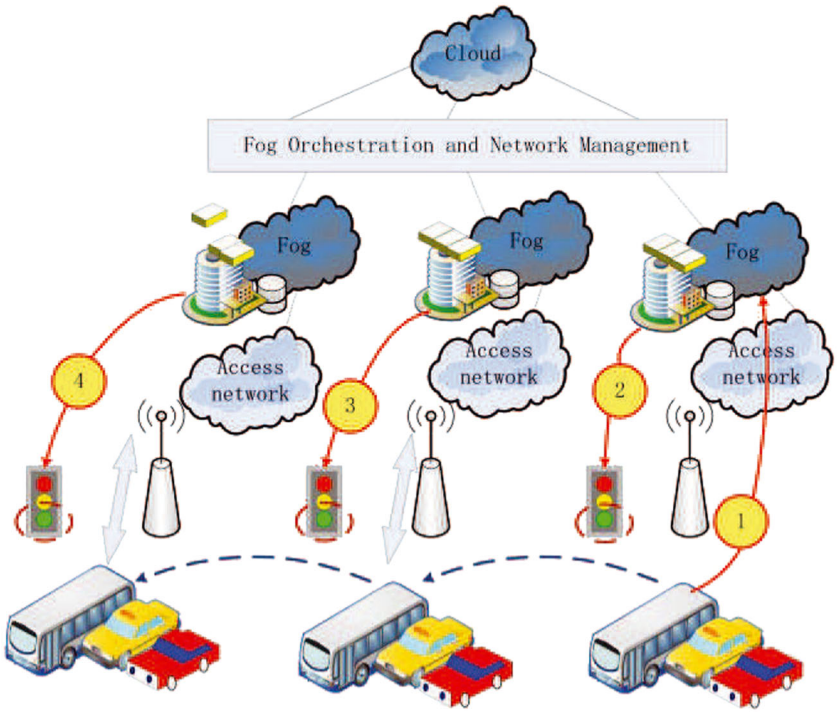

Figure 3. Fog computing in smart traffic lights and connected vehicles.

measure the distance and speed of approaching vehicles. As shown in Figure 3, intelligent lighting turns on once a sensor identifies movement and switches off as traffic passes. Neighbouring smart lights serving as Fog devices coordinate to create green traffic wave and send warning signals to approaching vehicles [2]. Wireless access points like WiFi, 3G, road-side units and smart traffic lights are deployed along the roads. Vehicle to vehicle, vehicle to access points and access points to access points interactions enrich the application of this scenario.

Wireless Sensor and Actuator Networks: Traditional wireless sensor networks fall short in applications that go beyond sensing and tracking, but require actuators to exert physical actions like opening, closing or even carrying sensors [2]. In this scenario, actuators serving as Fog devices can control the measurement process itself, the stability and the oscillatory behaviours by creating a closed-loop system. For example, in the scenario of self-maintaining trains, sensor monitoring on a train's ball-bearing can detect heat levels, allowing applications to send an automatic alert to the train operator to stop the train at next station for emergency maintenance and avoid potential derailment. In lifesaving air vents scenario, sensors on vents monitor air conditions flowing in and out of mines and automatically change air-flow if conditions become dangerous to miners.

Decentralized Smart Building Control: The applications of this scenario are facilitated by wireless sensors deployed to measure temperature, humidity or levels of various gases in the building atmosphere. In this case, information can be exchanged among all sensors in a floor, and their readings can be combined to form reliable measurements. Sensors will use distributed decision making and activation at Fog devices to react to data. The system components may then work together to lower the temperature, inject fresh air or open windows. Air conditioners can remove moisture from the air or increase the humidity. Sensors can also trace and react to movements (e.g. by turning the light on or off). Fog devices could be assigned at each floor and could collaborate on higher level of actuation. With Fog computing applied in this scenario, smart buildings can maintain their fabric, external and internal environments to conserve energy, water and other resources.

IoT and Cyber-physical systems (CPSs): Fog computing-based systems are becoming an important class of IoT and CPSs. Based on the traditional information carriers including Internet and telecommunication network, IoT is a network that can interconnect ordinary physical objects with

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

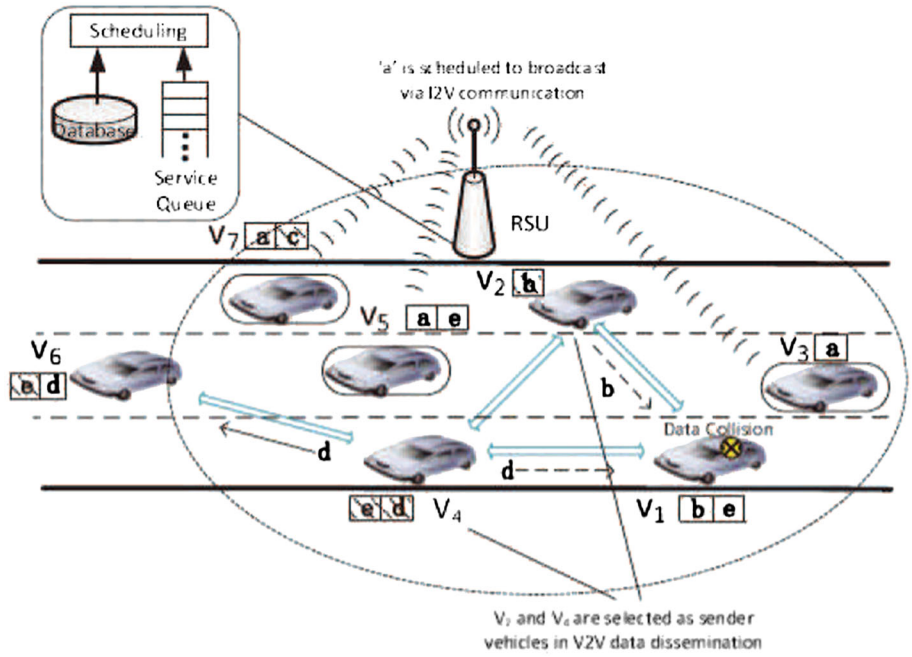

Figure 4. Fog computing in software defined network in vehicular networks [11]. RSU, roadside unit.

identified addresses [10]. CPSs feature a tight combination of the system's computational and physical elements. CPSs also coordinate the integration of computer and information centric physical and engineered systems. IoT and CPSs promise to transform our world with new relationships between computer-based control and communication systems, engineered systems and physical reality. Fog computing in this scenario is built on the concepts of embedded systems in which software programmes and computers are embedded in devices for reasons other than computation alone. Examples of the devices include toys, cars, medical devices and machinery. The goal is to integrate the abstractions and precision of software and networking with the dynamics, uncertainty and noise in the physical environment. Using the emerging knowledge, principles and methods of CPSs, we will be able to develop new generations of intelligent medical devices and systems, 'smart' highways, buildings, factories, agricultural and robotic systems.

Software Defined Networks (SDN): As shown in Figure 4, Fog computing framework can be applied to implement the SDN concept for vehicular networks. SDN is an emergent computing and networking paradigm and became one of the most popular topics in IT industry [12]. It separates control and data communication layers. Control is performed at a centralized server, and nodes follow communication path decided by the server. The centralized server may need distributed implementation. SDN concept was studied in wireless local area network (WLAN), wireless sensor and mesh networks, but they do not involve multi-hop wireless communication and multi-hop routing. Moreover, there is no communication between peers in this scenario. SDN concept together with Fog computing will resolve the main issues in vehicular networks, intermittent connectivity, collisions and high packet loss rate by augmenting vehicle to vehicle with vehicle to infrastructure communications and centralized control. SDN concept for vehicular networks is first proposed in [11].

4. STATE OF THE ART

A total of eight articles were identified in the concept of Fog computing [1, 2, 13–19]. There are some other concepts, not declared as Fog computing, fall under the same umbrella. We will also discuss these works in a subsection of the similar work.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

4.1. Related work

K. Hong *et al.* proposed mobile Fog in [14]. This is a high level programming model for geo-spatially distributed, large-scale and latency-sensitive future Internet applications. Following the logical structure shown in Figure 1, low-latency processing occurs near the edge, while latency-tolerant large-scope aggregation is performed on powerful resources in the core of the network (normally the Cloud). Mobile Fog consists of a set of event handlers and functions that an application can call. Mobile Fog model is not presented as generic model but is built for particular application while leaving out functions that deal with technical challenges of involved image processing primitives. Fog computing approach reduces latency and network traffic.

B. Ottenwalder *et al.* presented a placement and migration method for Cloud and Fog resources providers [18]. It ensures application-defined end-to-end latency restrictions and reduces the network utilization by planning the migration ahead of time. They also show how the application knowledge of the complex event processing system can be used to reduce the required bandwidth of virtual machines during their migration. Network intensive operators are placed on distributed Fog devices, while computationally intensive operators are in the Cloud. Migration costs are amortized by selecting migration targets that ensure a low expected network utilization for a sufficiently long time. This work does not optimize workload mobility because Fog devices are also able to carry computationally intensive tasks. It also does not optimize the size of control information or mobility overhead and does not describe network control policies for finding optimal paths for different applications.

In [16], K. Hong *et al.* proposed an opportunistic spatio-temporal event processing system that uses prediction-based continuous query handling. Their system predicts future query regions for moving consumers and starts processing events early so that the live situational information is available when the consumer reaches the future location. Historical events for a location are processed before the mobile user arrives at that location. Live event processing begins at the moment the user arrives. To mitigate large speed of mobile user, authors propose using parallel resources to enable pipeline processing of future locations in several time steps looking ahead. Further, they proposed taking several predictions for each time step and opportunistically compute the events for all of those locations. When the user arrives at that time, the prediction among those that is closest to truth will be selected and its events returned.

J. Zhu *et al.* applied existing methods for web optimization in a novel manner [19]. Within Fog computing context, these methods can be combined with unique knowledge that is only available at the Fog devices. More dynamic adaptation to the user's conditions can also be accomplished with network edge specific knowledge. As a result, a user's Web page rendering performance is improved beyond that achieved by simply applying those methods at the Web server.

In the mobile Cloud concept [17], pervasive mobile devices share their heterogeneous resources and support services. Neighbouring nodes in a local network form a group called a local Cloud. Nodes share their resources with other nodes in the same local Cloud. A local resource coordinator serving as Fog device is elected from the nodes in each local Cloud. The work in [17] proposed an architecture and mathematical framework for heterogeneous resource sharing based on the key idea of service-oriented utility functions. Normally, heterogeneous resources are quantified in disparate scales, such as power, bandwidth and latency. However, authors in [17] present a unified framework where all these quantities are equivalently mapped to 'time' resources. They formulate optimization problems for maximizing the sum and product of the utility functions and solve them via convex optimization approaches.

The work in [15] first reviews the reliability requirements of Smart Grid, Cloud, and sensors and actuators. This work then combines them towards reliable Fog computing. However, it only concludes that building Fog computing-based projects is challenging and does not offer any novel concept for the reliability of the network of smart devices in the Fog computing paradigm.

4.2. Similar work

BETaaS [20] proposed replacing Cloud as the resident for machine-to-machine applications by 'local Cloud' of gateways. The 'local Cloud' is composed of devices that provide smart things with

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

connectivity to the Internet, such as smart phones, home routers and road-side units. This enables applications that are limited in time and space to require simple and repetitive interactions. It also enables the applications to respond in a consistent manner.

Demand Response Management (DRM) is a key component in the smart grid to effectively reduce power generation costs and user bills. The work in [21] addressed the DRM problem in a network of multiple utility companies and consumers where every entity is concerned about maximizing its own benefit. In their model, utility companies communicate with each other, while users receive price information from utility companies and transmit their demand to them. They propose a Stackelberg game [22] between utility companies and end users to maximize the revenue of each utility company and the payoff of each user. Stackelberg equilibrium of the game has a unique solution. They develop a distributed algorithm that converges to the equilibrium with only local information available for both utility companies and end users. Utility companies play a non-cooperative game. They inform users whenever they change price, and users then update their demand vectors and inform utility companies. This iterates until convergence. The main drawback of this algorithm is a significant communication overhead between users and utility companies. Although DRM helps to facilitate the reliability of power supply, the smart grid can be susceptible to privacy and security issues because of communication links between the utility companies and the consumers. They study the impact of an attacker who can manipulate the price information from the utility companies and propose a scheme based on the concept of shared reserve power to improve the grid reliability and ensure its dependability.

The work in [23] investigated how energy consumption may be optimized by taking into consideration the interaction between both parties. The energy price model is a function of total energy consumption. The objective function optimizes the difference between the value and the cost of energy. The power supplier pulls consumers in a round-robin fashion and provides them with energy price parameter and current consumption summary vector. Each user then optimizes his own schedule and reports it to the supplier, which in turn updates its energy price parameter before pulling the next consumers. This interaction between the power company and its consumers is modelled through a two-step centralized game, based on which the work in [23] proposed the Game-Theoretic Energy Schedule (GTES) method. The objective of the GTES method is to reduce the peak to average power ratio by optimizing the user's energy schedules.

The closest work for SDN in vehicular networks are several implementations in wireless sensor network and mesh networks [24, 25]. Moreover, B. Zhou *et al.* studied adaptive traffic light control for smoothing vehicles' travel and maximizing the traffic throughout for both single and multiple lanes [26, 27]. In addition, the work in [28] proposed a three-tier structure for traffic light control. First, an electronic toll collection (ETC) system is employed for collecting road traffic flow data and calculating the recommended speed. Second, radio antennas are installed near the traffic lights. Third, road traffic flow information can be obtained by wireless communication between the antennas and ETC devices. A branch-and-bound-based real-time traffic light control algorithm is designed to smooth vehicles' travels.

5. SYSTEM SECURITY ON FOG DEVICES

5.1. System security issues

As a 'small cloud' close to the end users, Fog devices may encounter system security challenges because they are usually deployed in the places out of rigorous surveillance and protection. Traditional attacks become available to compromise the system of Fog devices in order to realize malicious aims such as eavesdropping and data hijack. In this section, we discuss the system security issues in Fog computing.

Intrusion detection techniques can also be applied in Fog computing [29]. Intrusion in smart grids can be detected using either a signature-based method in which the patterns of behaviour are observed and checked against an already existing database of possible misbehaviours. Intrusion can also be captured by using an anomaly-based method in which an observed behaviour is compared with expected behaviour to check if there is a deviation. The work in [30] develops an algorithm that

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

monitors power flow results and detects anomalies in the input values that could have been modified by attacks. The algorithm detects intrusion by using principal component analysis to separate power flow variability into regular and irregular subspaces.

5.2. An example: man-in-the-middle attack

The man-in-the-middle attack has a potential to become a typical attack in Fog computing. In this subsection, we take the man-in-the-middle attack as an example to expose the security problems in Fog computing. In this attack, gateways serving as Fog devices may be compromised or replaced by fake ones [6]. Examples are KFC or Star Bar customers connecting to malicious access points that provide deceptive service set identifier as public legitimate ones. Private communication of victims will be hijacked once the attackers take the control of gateways.

5.2.1. Environment settings of stealth test. Man-in-the-middle attack can be very stealthy in Fog computing paradigm. This type of attack will consume only a small amount of resources in Fog devices, such as negligible CPU utilization and memory consumption. Therefore, traditional anomaly detection methods can hardly expose man-in-the-middle attack without noticeable features of this attack collected from the Fog. In order to examine how stealthy the man-in-the-middle attack can be, we implement an attack environment shown in Figure 5. In this scenario, a 3G user sends a video call to a WLAN user. Because the man-in-the-middle attack requires to control the communication between the 3G user and the WLAN user, the key of this attack is to compromise the gateway that serves as the Fog device.

Two steps are needed to realize the man-in-the-middle attack for the stealth test. First, we need to compromise the gateway, and second, we insert malicious code into the compromised system. For susceptible gateways, we can either refresh the ROM of a normal gateway or place a fake active point in the environment. Both methods can be easily implemented in the real world, such as in the KFC or Star Bar environments. In our experiment, we choose the former and use Broadcom BCM5354 (Irvine, CA, USA) as the gateway [31]. This device has a high-performance MIPS32 processor (MIPS Technologies, Inc., Sunyvale, CA, USA), IEEE 802.11 b/g Media Access Controller/ Physical Layer and USB2.0 controller. Video communication is set up on BCM5354 between a 3G mobile phone and a laptop that adopts WiFi for connection. We refresh the ROM of BCM4354 and update its system to the open-source Linux kernel 2.4.

In order to hijack and replay victims' video communication, we insert a hook programme into the TCP/IP stack of the compromised system. Hook is a technique of inserting code into a system call in order to alter it [32]. The typical hook works by replacing the function pointer to the

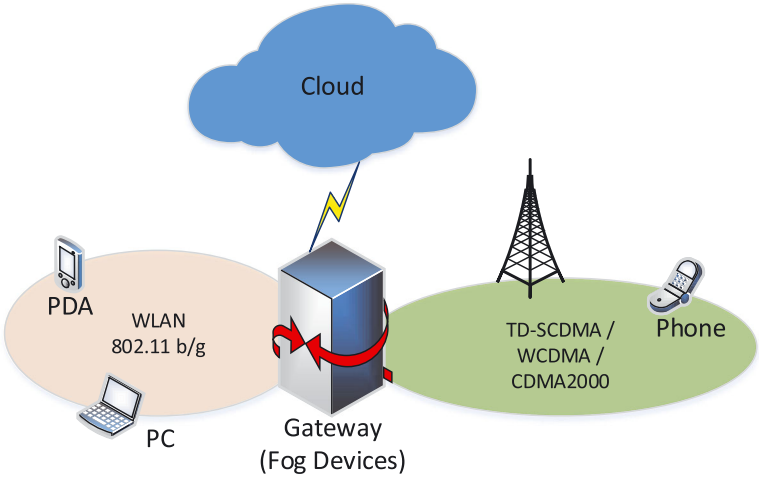

Figure 5. A scenario for a man-in-the-middle attack towards Fog. PDA, personal digital assistant; PC, personal computer; WLAN, wireless local area network; TD-SCDMA, time division synchronous code division multiple access; WC-DMA, wideband code division multiple access; CDMA, code division multiple access.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

call with its own, then once it is finished doing its processing, it will then call the original function pointer. The system structure is implemented in Figure 6. We further employ the relevant application programming interfaces and data structures in the system to control the gateway device, such as boot strap, diagnostics and initialization code. The IP packets from WLAN will be transferred to and processed in 3G related modules. We plug a 3G USB modem on BCM5354 device, on which we implement H.324M for video and audio tunnel with 3G circuit switched call setup. H.263 and Adaptive Multi-Rate functions are also implemented as the video and audio codec modules in the system.

5.2.2. Work flow of man-in-the-middle attack. The communication between 3G and WLAN needs a gateway to translate the data of different protocols into the suitable formats. Therefore, all the communication data will firstly arrive at the gateway and then be forwarded to other receivers.

In our experiment, the man-in-the-middle attack is divided into four steps. We illustrate the hijacked communication from 3G to WLAN in Figure 7. In the first two steps, the embedded hook process of the gateway redirects the data received from the 3G user to the attacker. The attacker replays or modifies the data of the communication at his or her own computer and then sends the

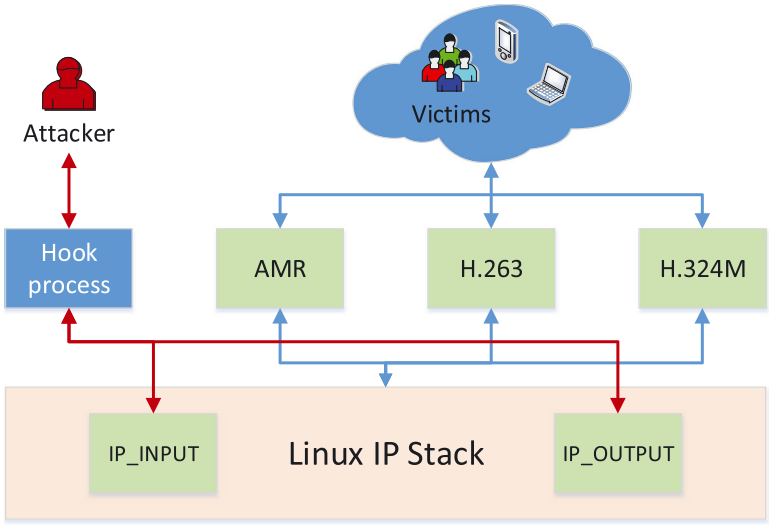

Figure 6. A system design of man-in-the-middle-attack in Fog. IP, Internet protocol, AMR, Adaptive Multi-Rate

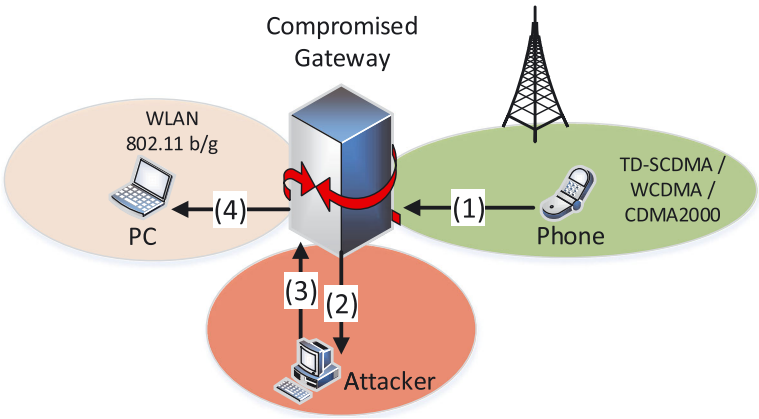

Figure 7. The hijacked communication in Fog (e.g. from phone to personal computer (PC)). WLAN, wireless local area network; TD-SCDMA, time division synchronous code division multiple access; WC-DMA, wideband code division multiple access; CDMA, code division multiple access.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

data back to the gateway. In the final step, the gateway forwards the data from the attacker to the WLAN user. In fact, the communication from the WLAN user will also be redirected to the attacker at first and then be forwarded by the hook in the gateway to the 3G user. We can see clearly from Figure 7 that the attacker can monitor and modify the data sent from the 3G user to the WLAN user in the 'middle' of the communication.

5.2.3. Results of stealth test. Traditional anomaly detection techniques rely on the deviation of current communication from the features of normal communication. These features include memory consumption, CPU utilization and bandwidth usage. Therefore, to study the stealth of man-in-the-middle attack, we examine the memory consumption and the CPU utilization of gateway during the attack. If man-in-the-middle attack does not greatly change the features of the communication, it can be proofed to be a stealthy attack. For simplicity, we assume the attacker will only replay the data at his or her own computer but will not modify the data.

Firstly, we compare the memory utilization of gateway before and after a video call tunnel is built in our experiment. The results are shown in Figure 8, and the red line in plots indicates the average amount of memory consumption. We can see clearly that man-in-the-middle attack does not largely influence the video communication. In Figure 8(A), the average value is 15 232 kb, while after we build the video tunnel on gateway, the memory consumption reaches 15 324.8 kb in Figure 8(B). Secondly, we show the CPU consumption of gateway in Figure 9. Based on the results in Figure 9, we can also see that man-in-the-middle attack does not largely influence the video communication. In the Figure 8(A), the average value is 16.6704%, while after the video tunnel is built, the CPU consumption reaches 17.9260%. We therefore conclude that man-in-the-middle attack can be very stealthy in Fog computing because of the negligible increases in both memory consumption and CPU utilization in our experiments.

Man-in-the-middle attack is simple to launch but difficult to be addressed. In the real world, it is difficult to protect Fog devices from compromise as the places for the deployment of Fog devices are normally out of religious surveillance. Encrypted communication techniques may also not protect users from this attack because attackers can set up a legitimate terminal and replay

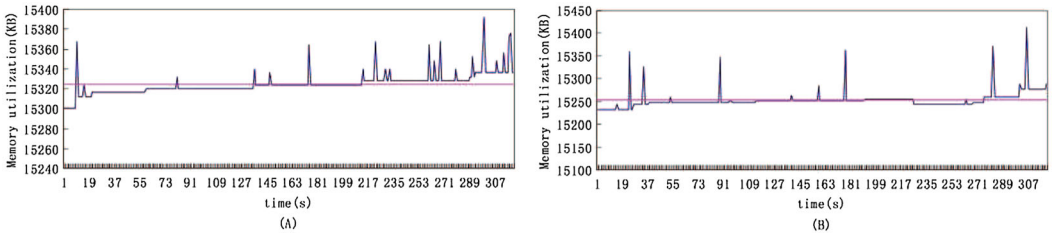

Figure 8. Memory Consuming of man-in-the-middle-attack in Fog.

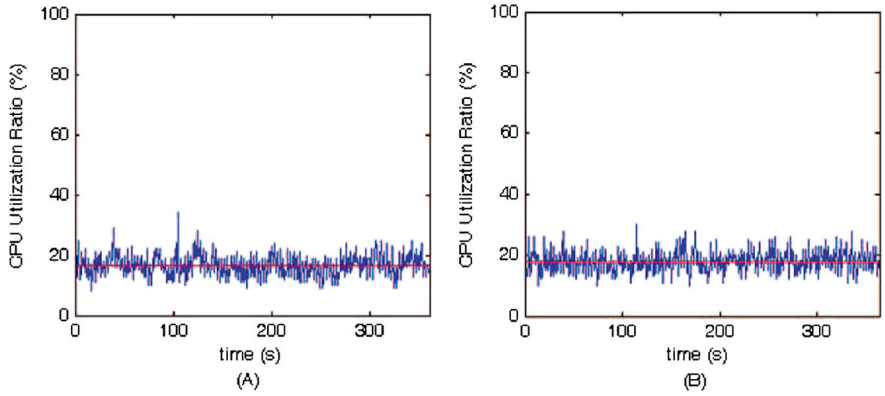

Figure 9. Central processing unit consuming of man-in-the-middle-attack in Fog.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

the communication without decryption. Particularly, complex encryption and decryption techniques may not be suitable for some scenarios. For example, the encryption and decryption techniques will consume lots of battery power in 3G mobile phones. In fact, this attack is not limited to the scenario of our experiment environment. We can find many applications running in Fog computing that are susceptible to man-in-the-middle attack. For example, many Internet users communicate with each other using MSN (Windows Live Massager). The communication data of MSN is normally not encrypted and can be modified in the 'middle'. Future work is needed to address the man-in-the-middle attack in Fog computing.

6. AUTHENTICATION AND AUTHORIZATION IN FOG COMPUTING

6.1. Authentication and authorization issues

Authentication and authorization issues were not studied in the context of Fog computing. They were studied in the context of smart grids [33] and machine-to-machine communications [34]. There are security solutions for Cloud computing. However, they may not suit for Fog computing because Fog devices work at the edge of networks. The working surroundings of Fog devices will face with many threats that do not exist in a well-managed Cloud. Different from system security on Fog devices, we discuss the security issues among Fog devices and between Fog and Cloud in this subsection.

Fog devices generally have some sort of connectivity to the remote Cloud authentication server that might be used to distribute authentication information and collect audit logs, but this connectivity may be as slow as 1200 baud in certain environments like smart grid [35]. As depicted in [35], performing an authentication protocol such as Remote Authentication Dial In User Service or Lightweight Directory Access Protocol over this connection is probably not desirable. Furthermore, reliance on Cloud central authentication servers is unwise because authentication should continue to apply for personnel accessing devices locally when remote authentication server communications are down. A provision to ensure that necessary access is available in emergency situations may be important, even if it means bypassing normal access control but with an audit trail.

This part of research also concerns with some privacy issues. In smart grids, privacy issues deal with hiding details, such as what appliance was used at what time, while allowing correct summary information for accurate charging. R. Lu *et al.* described an efficient and privacy-preserving aggregation scheme for smart grid communications [36]. It uses a super-increasing sequence to structure multi-dimensional data and encrypt the structured data by the homomorphic cryptogram technique. A homomorphic function takes as input the encrypted data from the smart metres and produces an encryption of the aggregated result. The Fog device cannot decrypt the readings from the smart metre and tamper with them. This ensures the privacy of the data collected by smart metres but does not guarantee that the Fog device transmits the correct report to the other gateways. For data communications from user to smart grid operation centre, data aggregation is performed directly on ciphertext at local gateways without decryption, and the aggregation result of the original data can be obtained at the operation centre [36]. Authentication cost is reduced by a batch verification technique.

6.2. An example: Authentication for fragile connection between Fog and Cloud

In this subsection, we take the fragile connection scenario between Fog and Cloud as an example to show the problems of the authentication and authorization in Fog computing. We will discuss a potential solution to this scenario. As shown in Figure 10, when the connection between Fog and Cloud is fragile, such as being broken, users would not be successfully authenticated as the authentication is deployed on Cloud server.

We introduced a new mechanism to realize user authentication when there is no connection to the Cloud server [37]. This kind of authentication is called Stand-Alone Authentication (hereinafter referred to as SAA).

We first introduce a normal authentication between user $\mathcal{C}$ and the Cloud server $\mathcal{AS}$:

$$\mathcal{C}[Credential] \overset{\text{Login}-\text{Auth}}{\Longleftrightarrow} \mathcal{AS}\,[SK_{\mathcal{AS}}, D_{\mathcal{AS}}] \rightarrow \{1, 0\}.$$

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

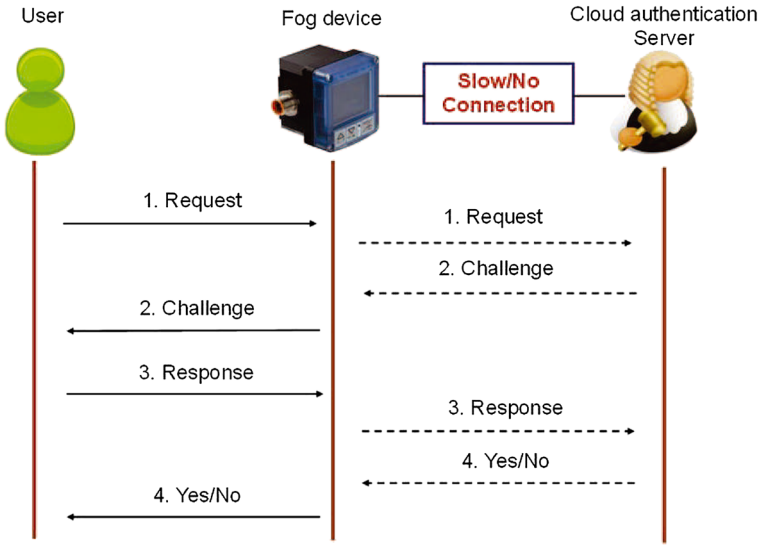

Figure 10. Authentication in Fog computing under fragile connections between Fog and Cloud.

At the server side, there are two kinds of information that may be necessary to authenticate a user: (1) the master secret key $SK_{\mathcal{AS}}$ and (2) the registration $D_{\mathcal{AS}}$ from the local database. It is evident that with $SK_{\mathcal{AS}}$ and $D_{\mathcal{AS}}$, anyone would be able to authenticate $\mathcal{C}$. Our design of SAA follows this idea but has the following differences.

First, the master key $SK_{\mathcal{AS}}$ should be known only by $\mathcal{AS}$ because of its critical role in the whole system. Thus, in our design, $\mathcal{AS}$ uses a randomly chosen key TK in each user registration and accordingly, TK will be used in authentication.

Second, taking the smart grid for example, it would not be practical to store $D_{\mathcal{AS}}$ on each device because of the large number of users and the limited storage space on intelligent electronic devices. Even in situations where storage is not an issue, maintaining and updating that information will be both time and effort consuming. Thus, in our design, $D_{\mathcal{AS}}$ is stored on smart-cards of users and sent to the Fog device during SAA.

Last, it would be certainly desirable if $\mathcal{AS}$ were able to determine which kind of Fog devices in the system can support SAA. In other words, TK and $D_{\mathcal{AS}}$ must be protected such that only the designated devices, that is, those that indeed need SAA, can obtain them. How to efficiently achieve such a kind of protection is the most challenging task in our design. As explained earlier, an extreme case we are concerned about is a newly joined user that must be authenticated by the Fog which however has lost the connection with $\mathcal{AS}$ since that user joined the system. The next two subsections give an insight on how to achieve such a control in a large-scale system.

6.2.1. Hybrid-encryption. A common approach to share data with a designated party is encryption: One can encrypt the data into a ciphertext such that only the intended party can decrypt and obtain the data. The process can use a secret key shared between two parties (known as symmetric-key encryption) or a public/private key pair of the designated party (known as public-key encryption).

From the operational point of view, public-key encryption is a desirable solution: Each device has a public key, a private key and a public-key certificate (if needed), and once installed, the private key never leaves the device. It does not require the level of coordination in symmetric-key settings, and the public key and its certificate can be publicly known. But public-key operations have a much higher computation requirement, and power-constrained devices cannot afford public-key operations very frequently.

In order to achieve a tradeoff between key management and efficiency, a commonly used approach is using hybrid-encryption: First, establish a secret key between communicating parties using public-key encryption, and then all following communications are secured using symmetric-key

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

encryption with the established key. Following this approach, $\mathcal{AS}$ can delegate the authentication right to a device as follows (high-level description):

- Choose a random Advanced Encryption Standard (AES) key key for each registration;
- Encrypt $(TK, D_{\mathcal{AS}})$ using AES and key: let C_1 be the output;
- Encrypt key using the device's public key: let C_2 be the output; and
- Store (C_1, C_2) on the smart card.

The designated device can retrieve key from C_2 (using its private key), and then decrypt C_1 (using key) to obtain $(TK, D_{\mathcal{AS}})$, which enables the Fog to authenticate the user when the connection to $\mathcal{AS}$ is down.

6.2.2. Attribute-based encryption. In an information system as large and dynamic as the smart grid, there are a large number of users and devices that need SAA. As an example, due to different roles, Alice can have SAA only with 'Fog devices in Area A OR Area B', but another user Bob can have stand-alone authentication with 'All Fog devices in Area A'. A naïve way to address this issue is extending the hybrid-encryption into a one-to-many setting.

In the approach described in Section 6.2.1, one can calculate C_2 as $\{C_2^1, C_2^2, \ldots, C_2^N\}$, where C_2^i is the encryption of key using the public key of the ith eligible device $\mathcal{D}_i$, that is, devices designated by the central authentication server. However, this naïve approach has two inherent drawbacks: (1) The size of C_2 increases linearly with the number of eligible devices and (2) Whenever there is a newly equipped device that needs stand-alone authentication, the server must calculate a new C_2^i and store it on each user's smart card. This apparently is not an easy task in a system with a large number of users. Overcoming these two drawbacks reminds us a recently introduced cryptographical primitive: Attribute-Based Encryption (ABE) [38].

Attributive-Based Encryption is a versatile tool for data provider, without prior knowledge of who exactly will be receiving the data, to share data with others in a more flexible way than a traditional end-to-end encryption. A typical example of ABE works as follows. Each potential data recipient is associated with an attribute set S and given a private key (generated by a third party) accordingly. The data provider can encrypt data with an embedded predicate function $f(\cdot)$, a description of which kind of recipients can decrypt the ciphertext correctly. Anyone with an attribute set S can successfully decrypt the ciphertext if and only if $f(S) = 1$.

Attributive-Based Encryption was first introduced in [38] under the name fuzzy identity-based encryption. Later on, several variants and improvements have been proposed [39–42]. Among them, the one that fits our situation is ciphertext-policy ABE (CP-ABE), where ciphertexts are associated with access policies, and keys are associated with sets of attributes. A CP-ABE consists of four algorithms: ABE.Setup, ABE.Enc, ABE.KeyGen and ABE.Dec.

This completed the description of the building blocks required by SAA. The detailed design can be found in [37]. By using the previously mentioned authentication method, users are able to be authenticated and authorized to Fog devices even when the connection between the Fog and Cloud is fragile.

7. CONCLUSIONS AND FUTURE WORK

We investigate Fog computing advantages for services in several domains and provide the analysis of the state of the art and security issues in current paradigm. Based on the work of this paper, some innovations in computation and storage may be inspired in the future to handle data intensive services based on the interplay between Fog and Cloud.

Future work will expand on the Fog computing paradigm in smart grid. In this scenario, two models for Fog devices can be developed. Independent Fog devices consult directly with the Cloud for periodic updates on price and demands, while interconnected Fog devices may consult each other and create coalitions for further enhancements.

Next, Fog computing-based SDN in vehicular networks will receive due attention. For instance, an optimal scheduling in one communication period, expanded towards all communication periods, has been elaborated in [11]. The Fog computing concept can also assist traffic light control.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

Finally, mobility between Fog nodes, and between Fog and Cloud, can be investigated. Unlike traditional data centres, Fog devices are geographically distributed over heterogeneous platforms. Service mobility across platforms needs to be optimized.

For the security issues, there is also some future work worth further investigation. For example, it is hard to avoid and defend the man-in-the-middle attacks. A promising solution would be to build an anti-tampering mechanism in the Fog device. There is another example: an online dictionary attacker in Fog computing makes authentication requests by trying every possible password for a specific user. In normal authentication, such attacks can be prevented using lockout mechanisms to lock out the user account after a certain number of invalid login attempts. However, the same approach does not apply to SAA in information systems with a large number of Fog devices: An attacker can run SAA with device D_1 using its guess PW_1, make another login request at device D_2 using another guess PW_2 and so on. This is like amounting online dictionary attacks on the same user in a distributed way. A naïve solution requires all devices sharing a common user list with failed login requests, but such a coordination would not be easily achievable in the situations need SAA (i.e. fragile communication environments). A satisfactory solution to thwart the distributed online dictionary attack is another direction of future work.

REFERENCES

1. Bonomi F. Connected vehicles, the internet of things, and Fog computing. *The Eighth ACM International Workshop on Vehicular Inter-Networking (VANET)*, Las Vegas, USA, 2011; 13–15.
2. Bonomi F, Milito R, Zhu J, Addepalli S. Fog computing and its role in the internet of things. *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, MCC'12, ACM, Ambleside, Unite Kingdom, 2012; 13–16.
3. Vaquero LM, Rodero-Merino L. Finding your way in the fog: towards a comprehensive definition of fog computing. *ACM SIGCOMM Computer Communication Review* 2014; **44**(5):27–32.
4. Hajibaba M, Gorgin S. A review on modern distributed computing paradigms: Cloud computing, jungle computing and fog computing. *CIT. Journal of Computing and Information Technology* 2014; **22**(2):69–84.
5. Manreet K, Monika B. Fog computing providing data security: a review. *International Journal of Advanced Research in Computer Science and Software Engineering* 2014; **4**(6):832–834.
6. Zhang L, Jia W, Wen S, Yao D. A man-in-the-middle attack on 3G-WLAN interworking. *International Conference on Communications and Mobile Computing (CMC)*, Vol. 1, Zhangjiajie, China, April 2010; 121–125.
7. Stojmenovic I, Wen S. The Fog computing paradigm: scenarios and security issues. *2014 Federated Conference on Computer Science and Information Systems (FeDCSIS)*: IEEE, Warsaw, Poland, 2014; 1–8.
8. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. A view of cloud computing. *Communications of the ACM* 2010; **53**(4):50–58.
9. Wei C, Fadlullah Z, Kato N, Stojmenovic I. On optimally reducing power loss in micro-grids with power storage devices. *IEEE Journal of Selected Areas in Communications* 2014; **32**(7):1361–1370.
10. Atzori L, Iera A, Morabito G. The internet of things: a survey. *Computer Networks* 2010; **54**(15):2787–2805.
11. Liu K, Ng J, Lee V, Son S, Stojmenovic I. Cooperative data dissemination in hybrid vehicular networks: Vanet as a software defined network, 2014. *Submitted for publication*.
12. Kirkpatrick K. Software-defined networking. *Communication of the ACM* 2013; **56**(9):16–19.
13. Cisco. Cisco delivers vision of fog computing to accelerate value from billions of connected devices, Cisco, January 2014.
14. Hong K, Lillethun D, Ramachandran U, Ottenwälder B, Koldehofe B. Opportunistic spatio-temporal event processing for mobile situation awareness. *Proceedings of the 7th ACM International Conference on Distributed Event-Based Systems*, DEBS'13, ACM, Arlington, TX, USA, 2013; 195–206.
15. Madsen H, Albeanu G, Burtschy B, Popentiu-Vladicescu FL. Reliability in the utility computing era: towards reliable fog computing. *2013 20th International Conference on Systems, Signals and Image Processing (IWSSIP)*, Bucharest, Romania, July 2013; 43–46.
16. Hong K, Lillethun D, Ramachandran U, Ottenwälder B, Koldehofe B. Mobile fog: a programming model for large-scale applications on the internet of things. *Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing*, MCC'13, ACM, Hongkong, 2013; 15–20.
17. Nishio T, Shinkuma R, Takahashi T, Mandayam NB. Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud. *Proceedings of the First International Workshop on Mobile Cloud Computing and Networking*, MobileCloud'13, ACM, Bangalore, India, 2013; 19–26.
18. Ottenwalder B, Koldehofe B, Rothermel K, Ramachandran U. MigCEP: operator migration for mobility driven distributed complex event processing. *Proceedings of the 7th ACM International Conference on Distributed Event-Based Systems*, DEBS'13, ACM, Arlington, TX, USA, 2013; 183–194.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe

19. Zhu J, Chan D, Prabhu M, Natarajan P, Hu H, Bonomi F. Improving web sites performance using edge servers in fog computing architecture. *2013 IEEE 7th International Symposium on Service Oriented System Engineering (SOSE)*, Wailea-makena, HI, USA, March 2013; 320–323.
20. BETaaS. Building the environment for the things as a service, BETaaS, Nov. 2012.
21. Maharjan S, Zhu Q, Zhang Y, Gjessing S, Basar T. Dependable demand response management in the smart grid: a stackelberg game approach. *IEEE Transactions on Smart Grid* March 2013; **4**(1):120–132.
22. Korzhyk D, Conitzer V, Parr R. Solving Stackelberg games with uncertain observability. *The 10th International Conference on Autonomous Agents and Multiagent Systems - volume 3*, AAMAS '11, Taipei, Taiwan, 2011; 1013–1020.
23. Fadlullah Z, Quan D, Kato N, Stojmenovic I. GTES: an optimized game-theoretic demand-side management scheme for smart grid. *IEEE Systems Journal* 2014; **8**(2):588–597.
24. Luo T, Tan HP, Quek T. Sensor openflow: enabling software-defined wireless sensor networks. *IEEE Communications Letters* 2012; **16**(11):1896–1899.
25. Daraghmi Y, Yi CW, Stojmenovic I. Forwarding methods in data dissemination and routing protocols for vehicular ad hoc networks. *IEEE Network* 2013; **27**(6):74–79.
26. Zhou B, Cao J, Zeng X, Wu H. Adaptive traffic light control in wireless sensor network-based intelligent transportation system. *2010 IEEE 72nd Vehicular Technology Conference Fall (VTC 2010-Fall)*, Ottawa, Canada, September 2010; 1–5.
27. Zhou B, Cao J, Wu H. Adaptive traffic light control of multiple intersections in wsn-based its. *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, Budapest, Hungary, May 2011; 1–5.
28. Li C, Shimamoto S. An open traffic light control model for reducing vehicles CO2 emissions based on etc vehicles. *IEEE Transactions on Vehicular Technology* January 2012; **61**(1):97–110.
29. Modi C, Patel D, Borisaniya B, Patel H, Patel A, Rajarajan M. A survey of intrusion detection techniques in Cloud. *Journal of Network and Computer Applications* 2013; **36**(1):42–57.
30. Valenzuela J, Wang J, Bissinger N. Real-time intrusion detection in power system operations. *IEEE Transactions on Power Systems* 2013; **28**(2):1052–1062.
31. Broadcom BCM 5354. (Available from: http://www.broadcom.com.) [Accessed on 2 April 2015].
32. Wikipedia. Hooking, what is hooking? 2014. (Available from: http://en.wikipedia.org/wiki/Hooking) [Accessed on 2 April 2015].
33. Wang W, Lu Z. Survey cyber security in the smart grid: survey and challenges. *Computer Networks* 2013; **57**(5):1344–1371.
34. Lu R, Li X, Liang X, Shen X, Lin X. GRS: the green, reliability, and security of emerging machine to machine communications. *IEEE Communications Magazine* 2011; **49**(4):28–35.
35. NIST. Guidelines for smart grid cyber security (NIST 7628), 2010. (Available from: Http://csrc.nist.gov/publications/PubsNISTIRs.html) [Accessed on 2 April 2015].
36. Lu R, Liang X, Li X, Lin X, Shen X. EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications. *IEEE Transactions on Parallel and Distributed Systems* 2012; **23**(9):1621–1631.
37. Huang X, Xiang Y, Bertino E, Zhou J, Xu L. Robust multi-factor authentication for fragile communications. *IEEE Transactions on Dependable and Secure Computing* 2014; **11**(6):568–581.
38. Sahai A, Waters B. Fuzzy identity-based encryption. *Eurocrypt*, Aarhus, Denmark, 2005; 457–473.
39. Lewko AB, Okamoto T, Sahai A, Takashima K, Waters B. Fully secure functional encryption: attribute-based encryption and (hierarchical) inner product encryption. *Eurocrypt*, Riviera, French, 2010; 62–91.
40. Lewko AB, Waters B. Unbounded HIBE and attribute-based encryption. *Eurocrypt*, Tallinn, Estonia, 2011; 547–567.
41. Waters B. Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. *Public Key Cryptography*, Taormina, Italy, 2011; 53–70.
42. Ostrovsky R, Sahai A, Waters B. Attribute-based encryption with non-monotonic access structures. *ACM Conference on Computer and Communications Security*, Alexandria, VA, USA, 2007; 195–203.

Copyright © 2015 John Wiley & Sons, Ltd.

DOI: 10.1002/cpe