Baker, T, Asim, M, Mac Dermott, AM, Iqbal, F, Kamoun, F, Shah, B, Alfandi, O and Hammoudeh, M

A Secure Fog-based Platform for SCADA-based IoT Critical Infrastructure

A Secure Fog-based Platform for SCADA-based IoT Critical Infrastructure

Thar Baker[1], Muhammad Asim[2], Áine MacDermott[1], Farkhund Iqbal[3], Faouzi Kamoun[4], Babar Shah[3], Omar Alfandi[3], Mohammad Hammoudeh[5]

[1] Department of Computer Science, Liverpool John Moores University, UK; {t.baker, a.m.macdermott}@ljmu.ac.uk
[2] Department of Computer Science, National University of Computer and Emerging Sciences, Pakistan; Muhammad.asim@nu.edu.pk
[3] College of Technological Innovation, Zayed University, UAE; {farkhund.iqbal, Babar.shah, Omar.alfandi}@zu.ac.ae
[4] Esprit School of Engineering, Tunisia; faouzi.kammoun@esprit.tn
[5] School of Computing, Math and Digital Technology, Manchester Metropolitan University, UK; m.hammoudeh@mmu.ac.uk

Abstract: The rapid proliferation of Internet of Things (IoT) devices, such as smart meters and water valves, into industrial critical infrastructures and control systems has put stringent performance and scalability requirements on modern Supervisory Control and Data Acquisition (SCADA) systems. While cloud computing has enabled modern SCADA systems to cope with the increasing amount of data generated by sensors, actuators and control devices, there has been a growing interest recently to deploy edge datacenters in fog architectures to secure low-latency and enhanced security for mission-critical data. However, fog security and privacy for SCADA-based IoT critical infrastructures remains an under-researched area. To address this challenge, this contribution proposes a novel security *"toolbox"* to reinforce the integrity, security, and privacy of SCADA-based IoTcritical infrastructure at the fog layer. The toolbox incorporates a key feature: a cryptographic-based access approach to the cloud services using identity-based cryptography and signature schemes at the fog layer. We present the implementation details of a prototype for our proposed Secure Fog-based Platform (SeFoP) and provide performance evaluation results to demonstrate the appropriateness of the proposed platform in a real-world scenario. These results can pave the way towards the development of more secured and trusted SCADA-based IoT critical infrastructure, which is essential to counter cyber threats against next-generation critical infrastructure and industrial control systems. The results from the experiments demonstrate a superior performance of SeFoP, which is around 2.8 seconds when adding 5 virtual machines (VMs), 3.2 seconds when adding 10 VMs, and 112 seconds when adding 1000 VMs compared to Multi-Level user Access Control (MLAC) platform.

Keywords: Fog computing; Secured platform; IoT; SCADA systems; Critical Infrastructure; mission-critical data.

1. Introduction

The adoption of cloud computing to support business operations has increased remarkably due to the on-demand pricing model; which has led to reducing the Total Cost of Ownership (TCO) for private data centers. Despite the increasing usage of cloud computing for dynamic provisioning of resources and pay-per-use on demand access, there are still unresolved issues such as the requirement of high capacity client access links, variable latency, lack of mobility support, security concerns and location-awareness [1][2]. In particular, applications such as real-time monitoring industrial automation (Industry 4.0), sensor and actuators networks, and intelligent transportation systems are too latency-sensitive to be deployed on cloud infrastructures. Generally, cloud data centers are located near the core network and in places where operating cost is lowest. Thus, applications and services might suffer unacceptable round-trip latency, when data is transmitted

from/to end devices, to/from the remote cloud data center through multiple gateways. Fog computing is an emerging paradigm that can address this challenge by offering computing resources and services at the edge of the network. Fog computing puts a substantial amount of communication, control, storage and management at the edge of a network as opposed to establishing dedicated channels to a more centralized remote cloud infrastructure. This approach reduces service latency, improves quality of service and provides a superior experience to end-users [2][3].

Supervisory Control and Data Acquisition (SCADA) systems are widely used for the control, monitoring and automation of industrial processes and Critical Infrastructure (CI) systems. Centralized SCADA systems were first introduced in 1960 for CI monitoring and control purposes [4]. The use of SCADA technology has evolved and is expected to grow substantially up from €188 million in 2007 towards €300 million in 2020 [5]. These systems were vendor-controlled and strictly isolated from other systems [6][7][8] focusing specially on collecting the necessary log-field data using industrial Distributed Control Systems (DCS), which recorded the data and sent every reading to the control room to (re)act accordingly. In case of emergency, a DCS message is generated and is raised as a critical alarm to inform the control room operator to react to the situation accordingly. CIs are vulnerable to cyber-attacks and emerge from a wide range of prospective perpetrators: state-sponsored terrorism, espionage and sabotage, malevolent ''hacktivists,'' or even disgruntled insiders. Cyber-attacks against SCADA-based IoT CIs are fueled by the high value and intrinsic vulnerabilities of these infrastructures, and this has led governments and security agencies to look for more effective ways to secure them [9]. For example, in the United Kingdom, the Centre for the Protection of National Infrastructure (CPNI) has been issuing security guidance on protecting CI facilities [8]. Though IoT inherits the same monitoring requirements from Cloud computing, the related challenges are however further amplified by the volume, variety, and velocity characteristics of IoT data.

It has been observed that the protection of CIs against cyber-attacks hinges on the security of the associated ICT infrastructure. One notable instance was the 2010 cyber-attack on Iran's Natanz nuclear facility through the highly modified Stuxnet malware that was responsible for causing substantial damage [10]. In another example, the HAVEX malware targeted industrial control systems by attaching itself to software updates distributed by control system manufacturers. When downloaded, it collects information from control devices and sends the information back to the remote attacker for analysis [11]. Thus, an increasing number of research initiatives are actively seeking advanced security mechanisms to protect CI assets and networks. For example, the PROTECT Centre [12], established in 2011, conducts research into critical infrastructure computer technology and protection. The security solutions for CI go beyond the traditional deployment of firewalls and Intrusion Detection Systems (IDS). They include strong user authentication, Demilitarized Zones (DMZs), system and protocol hardening, and trusted systems, among many others [13]. Since each service requires various conflicting types of security objectives such as authorization and authentication, current models often fail to provide an effectively abstracted security layer to accommodate the different authorization requirements for each service. Hence further research is warranted to investigate how to effectively integrate security and role-based authentication support [14]. The IoT is becoming a major source of big data, generating huge amounts of streamed information from many interconnected nodes, which have to be stored, processed, and presented in an efficient and easily interpretable fashion [15]. The amount of data generated by the number of IoT devices integrated into industrial control systems is expanding at an ever-increasing rate; which means that modern SCADA systems must have the ability to manage thousands of times more data than traditional SCADA systems [16]. For example, in large smart grids, the amount of data that is generated by the geographically distributed smart meters is so massive that moderate cloud computing resources cannot cope with. In this regard, Cisco determined that "today's cloud models are not designed for the volume, variety, and velocity of data that IoT generates." A direct upload of all data generated from IoT devices to the cloud for storage, processing and analysis would need unfathomable measures of data transfer capacity. There exist various studies that aim to address

cloud computing issues for industrial IoT-Based SCADA systems [17][18][19][20]. Moreover, private customer data originating from ubiquitous devices (such as smart phones) are usually stored in cloud infrastructures; which not only attract intruders but also stakeholders, such as service providers and cloud operators who are interested to use this data for their own benefits (e.g. advertising) [21][22]. The Federal Trade Commission (FTC) Report [23] on IoT urged businesses to adopt best practices to address consumer privacy concerns and security risks. It warns that smart devices are involved in harvesting huge amount of personal information and that they are exposed to a variety of potential security threats, such as unauthorized access and misuse of personal information.

Fog computing can help to address some of the aforementioned issues in modern cloud-based SCADA systems [2][4][17][19][20]. For instance, Fog computing facilitates the on-site data storage and analysis of time-sensitive heterogeneous data by reducing the amount of data stored and transmitted to the cloud; thus, providing better delay performance. However, Fog computing is a non-trivial extension of cloud computing, and it has been proposed in the context of the IoT. Thus, most cloud computing security and privacy issues are inherited in fog computing. Also, due to the underlying differences between cloud computing and fog computing, security solutions proposed for cloud computing may not suit fog services which are available to end users at the edge of networks. This might hinder the integration of Fog paradigm into the SCADA-based IoT Critical Infrastructure.

In this paper we present a novel end-to-end security approach as a part of a security toolbox to reinforce the integrity, security, and privacy of SCADA-based IoT Critical Infrastructure at the fog layer. The initial design and key aspects of the '*security toolbox*' in the context of a centralized SOA-based SCADA systems have been described in a previous work [14], whereby the authors proposed an open platform enabling the integration of cloud computing into SOA based SCADA systems. The contribution in [14] also identified the key elements required to enforce security and integrity as a SOA-based service on these systems. This paper amends and extends the security toolbox to reinforce the integrity, security, and privacy of IoT-based SCADA systems at the fog layer.

The present contribution differs from previous work through the following two main aspects:

(i) The proposal of a cryptographic-based access approach for the cloud services using identity-based cryptography and identity-based signature schemes at the fog layer. To our knowledge, this is the first contribution that aimed to explore the usage of identity-based cryptography and identity-based signature schemes within the fog layer.

(ii) Implementation and evaluation of the proposed Secure Fog-based Platform (SeFoP)

The rest of the paper is structured as follows: the next section presents the motivations behind this work and highlights the research problems that this contribution aims to address. Section 3 provides a literature survey and a summary of earlier contributions. Section 4 presents our approach to secure a SCADA-based IoT critical infrastructure, and Section 5 details our proposed fog-based cryptographic solution using identity-based cryptography and identity-based signature. Section 6 outlines the implementation details of the proposed key authentication mechanism while Section 7 presents and discusses preliminary evaluation results. Finally, Section 8 concludes the paper and indicates the direction of our future work.

2. Research Motivations and Problem Statement

2.1. Research Motivations

Modern SCADA systems are adopting the emerging IoT paradigm and commercial cloud computing services for the monitoring and protection of critical applications associated with industrial control systems, smart grids, and other CIs [24][25]. As a result, the threats posed by the expansion of the

cyber vulnerability landscape have increased dramatically in terms of potential, frequency, and impact [7].

In smart industrial systems such as manufacturing, oil and gas, eHealth systems, and transportation, fast response-time is crucial to improve service level and increase safety. In fact, a few milliseconds matter when it comes to shutting down a power plant or restoring electrical services. Analyzing extremely latency-sensitive data close to the smart objects that collected the data is one of the merits of Fog computing. For SCADA-based IoT critical infrastructures, this paradigm can help in providing better delay performance and can potentially make a difference between averting an imminent disaster and assuming the consequences of a cascading system failure. However, by embracing the fog computing paradigm, fog-enabled IoT-based SCADA systems become the subject of a new breed of cyber threats and attacks. This work is motivated by the fact that addressing security and privacy concerns at the fog layer could enable fog computing to provide not only additional computational resources, but also adequate level of security to thwart cyber-attacks against SCADA-based IoT critical infrastructures. At the same time, we acknowledge that special care should be exercised to ensure that any proposed security/privacy mechanism should not compromise system performance and scalability requirements. Failure to do so will negate the gain brought by fog computing in terms of lower response time and latency in support for real-time SCADA-based IoT critical services.

2.2. Problem Statement

Lee et al. [26] suggested that in the near future, IoT fog computing is expected to collect and process deeply personal information originated from millions of IoT devices. While existing security solutions can be used to address some threats, there are other issues that pertain to fog computing environments and which pose unique challenges for security researchers and practitioners [21]. For example, authentication is an essential requirement for protecting IoT data both in transit and at rest. Unfortunately, many IoT devices are not equipped with enough memory and CPU power to perform cryptographic operations that are required by most of today's authentication protocols. It is however possible to migrate the expensive processing and storage from the IoT device to the fog layer. For example, a security toolbox can be deployed as a service over the fog computing platform to enable the resource-constrained IoT devices execute their authentication service and make use of other security services. This however opens new attack vectors against the fog computing platform such as fog node-compromised attack [26]. In summary, without deploying proper security and privacy-preserving mechanisms, the adoption and diffusion of fog computing in IoT-based SCADA systems will remain restricted. Accordingly, this contribution aims to address the following research question: What mechanisms can be put in place to effectively enhance the security and privacy of cloud-assisted IoT-based SCADA systems, while not compromising performance and scalability requirements?

3. Related Work

Modern industrial cyber physical systems often depend on SCADA systems to control and monitor their infrastructure. They are considered smart industrial systems due to their association with IoT, machine learning, artificial intelligence and Cloud computing [19][27]. Moreover, IoT is becoming a main source of big data, generating an enormous amount of information, flowing through many connected nodes. This information has to be stored, processed, and presented in an effective, efficient, and easily interpretable form. In principle, fog services can be positioned between cloud-based SCADA systems and the cloud to enable a wide range of benefits, such as enhanced bandwidth utilization, reduced latency, enhanced security, heterogeneity and geophysical distribution. However, research on integrating fog computing into IoT-based SCADA systems is still in its infancy. We divide the study of existing literature into a fog-based and SCADA-based solutions.

3.1 Fog-based solutions

Recent work has shown how fog complements and extends cloud computing, emphasizing fog's relevance to several verticals within the IoT and Big Data space [28][29][30][31]. Bonami et al. [30] presented a high-level description of fog software architecture, articulating the different technological components necessary to implement the fog paradigm. Hong et al. [32] proposed the concept of mobile fog, a high-level programing model to support latency-sensitive and large-scale IoT applications that are geospatially distributed. Do et al., [33] analyzed the joint resource allocation and carbon footprint problem in fog computing. Authors in [34] proposed a conceptual model to tackle the issues of resource prediction, resource estimation and reservation, and pricing for new and existing IoT customers. Dubey et al. [35] proposed a service-oriented architecture based on fog computing to perform data mining and analysis on raw data collected from various wearable sensors used for telehealth applications. Zao et al. [36] developed a brain computer interaction application using fog computing where data is pre-processed on the fog aiming to reduce latency and bandwidth utilization. The authors in [17] explored fog computing capabilities for smart grids. They proposed a reference model that integrates fog computing concepts into smart grids.

3.2 SCADA-based solutions

Classical SCADA systems are already lacking proper security and privacy preserving mechanisms. However, due to their integration with new complex architectures, enabled by the cloud computing paradigm and IoT, they are becoming the subject of new security threats. Each point in the SCADA network is a potential entry point –this is in part due to the fact that little work has been done to enhance its security because of the misconception that such a network is inherently secured. The connectivity of SCADA networks increases the risk of cyber-attacks and hence there is a need to improve the security of these networks. SCADA systems are rarely patched or updated as engineers are often hesitant to do so due to concerns that the patch itself could potentially adversely impact the operation of the system [37]. The authors in [27] highlighted the security challenges of cloud-assisted IoT-based SCADA systems and provided recommendations and best practices for improving and maintaining their security. MacDermott et al. [37] investigated the factors that need to be taken into consideration when CI services are hosted in a cloud environment. The authors detailed some security concerns and existing protection methods. Piggin [38] discussed the opportunities and risks associated with the migration of SCADA and Industrial Control Systems (ICS) to cloud-based environments. In CI systems, the introduction of the six Trusted Computing elements [39] into cloud computing could possibly help to further strengthen the security of these systems. This is still seen as a new undeveloped solution for secure computing, albeit several research studies have already aimed at designing and developing trusted cloud computing services [40]. One (obvious) area where trusted cloud computing can be applied is in the protection of the underlying infrastructure, including the data centers and interconnected networks. The deployment of encrypted data storage, memory curtaining, and protected execution environments, based on some particular form of the trusted platform module (TPM) architecture, could make a substantial contribution in isolating and securing cloud resources in virtualized environments [40].

In a typical SCADA-based IoT infrastructure, an increasing number of connected smart devices, with potential security vulnerabilities, can request communication with cloud resources for data storage and retrieval. As mentioned previously, the IoT inherits the same monitoring requirements from cloud computing, but the related challenges are further affected by volume, variety, and velocity characteristics of IoT data. One single weak link in the security chain could provide attackers with doorways that could potentially be unlocked and lead to data breach [41][42]. Alrawais et al. [43] discussed the security and privacy issues in IoT environments and proposed a method that employs fog computing to improve the distribution of certificate revocation information among IoT devices for security enhancement. Dsouza et al. [1] proposed a policy-driven security management approach for fog resources including policy analysis and its integration with the fog paradigm. However, the approach fails to consider rights inheritance or propagation of rights in the ecosystem [44][45]. Stojmenovic et al. [45] presented a survey of fog applications and the associated security

challenges. In particular, the authors analyzed man-in-the-middle attacks in the context of fog computing where fog devices can be compromised or replaced by fake ones. This paper takes one step further to explore the design and implementation aspects of practical solutions to enhance the security and privacy of cloud-enabled IoT-based SCADA systems at the fog level. This will enable the fog paradigm to offer not only a balanced mix of enhanced computation power and connectivity but also provide an adequate level of security to counter cyber-attacks.

4. An Integrated Approach to Secure cloud-enabled IoT-based SCADA CI

The integration of SCADA and CI systems into IoT infrastructures gave rise to new security challenges that warrant further investigation. These challenges motivated us to explore a new approach towards an integrated secure fog platform that aims to address most of the security concerns in cloud-enabled IoT-based SCADA critical infrastructures. In the subsequent sections, we present in detail our integrated approach. We first analyse the security and performance requirements of SCADA CI systems and then outline the main features and functionalities of the proposed platform.

4.1. Critical Infrastructure Requirements

A major concern for moving CI data into a cloud computing environment relates to the stringent requirement for high security, low latency/quick response and high service availability. Whilst it is unlikely that CI providers will migrate their mission-critical services to public cloud environments, support systems and tertiary services, on the other hand, may be good candidate for cloud hosting. In this case, the main requirements involve: (a) real time support to guarantee a high level of availability in case of faults or intermittent connectivity problems; (b) scalability to cope with large volumes of data transfer at variable rates without compromising performance; (c) infrastructure security to protect data both at rest in the cloud and in transit; (d) high availability (e.g., reliability and resilience to reduce interruption); (e) least charges of transitioning and maintaining the cloud service; (f) dynamic provisioning to adapt to varying workloads and to cope with spikes and flash crowds; and (g) legal assurances that the customer can specify and receive a fine degree of control over the service hosting and the data replication strategy as stipulated by the service level agreement (SLA).

While many of the abovementioned requirements can be convened by cloud computing solutions, there are several well-known issues that are potentially precarious with CIs. Among the most critical concerns is the relative lack of strong security and user authentication in typical cloud platforms and the limited control and monitoring of data replication and service location inside the cloud. A data-centric security approach must ensure that data protection mechanisms are deployed across all provided security solutions, and that data owners have the full control over authentication and authorization rights and privileges. Institutional security policies and access rules can be specified and mapped to the cloud environment. Requirement based security issues can be different for CI applications and for common IT applications but need to be considered in combination for the given context [46]. With this in mind, our proposed platform is detailed below.

4.2. Platform Features and Functionality

The main feature of the proposed platform is supporting the migration of SCADA services to a secured IoT-cloud platform via a new secured fog computing layer. The assumption used in the proposed approach is that all the connected systems in the cloud environment provide their functionalities (integrated or atomic) as-a-service that can be further composed by and interacted with other systems' components subscribed to the cloud. The access to these services and communications (i.e., sending and receiving signals/data) will be enabled by the new added, secured fog layer. The platform will reinforce data integrity and security to minimize the risk that mission critical services will be interrupted or affected by security attacks. In addition, the platform alleviates the need to check the validity of sending/receiving certificates, which slows down the communication

process noticeably in an IoT environment due to the huge amount of frequent communication. We do so by encrypting the certificates at the fog layer, hence there is no need to check for the validity of these encrypted certificates every time they are in use.

To achieve our objectives, we propose three key services at the fog layer, *Planning services*, *End-to-End Security services*, and *Monitoring and Policing services*. These services will be presented in the context of a *'security toolbox'* that will be hosted at the fog layer and allows the platform to be adopted and deployed as main components by a range of cloud providers and users to allow secured communication, as illustrated in Figure 1. However, this paper focuses mainly on the *End-to-End Security services* with the encrypted access approach to the cloud services using identity-based cryptography and identity-based signature.

Specialized unified threat management (UTM) systems will be deployed in order to secure the cloud against potential attacks and this will be complemented with resilient networking services, based on Software Defined Networking (SDN) mechanisms to counter recognized attack patterns. For the purpose of *end-to-end security*, we exploit two cryptographic–schemes to encrypt the user communications and messages to the cloud. The main advantage is the provision of a trusted and secured public key generator, which will be part of our security toolbox, at the fog layer. This can potentially upsurge the security rigidity and reduce key management burden by eliminating the traditional need for a trusted third party to generate a public key.

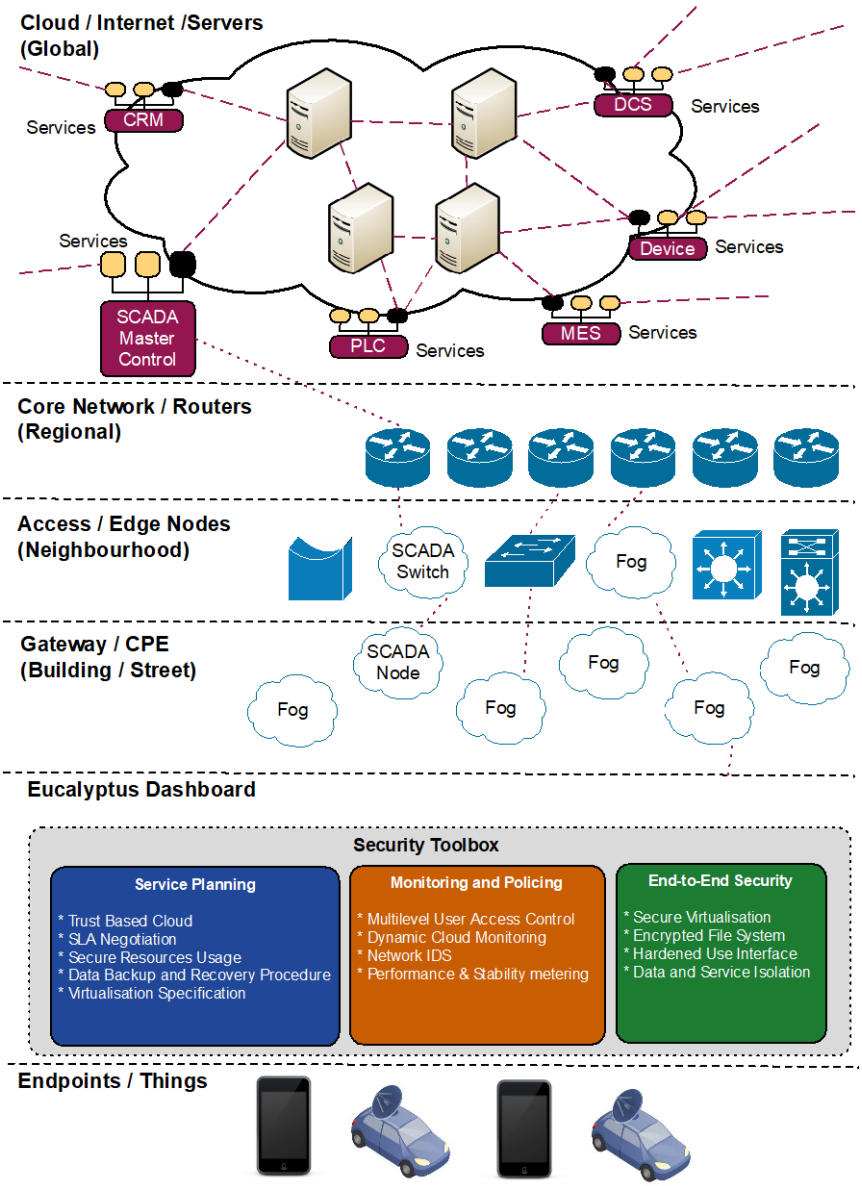

Figure 1: High-level overview of the proposed fog-based Toolbox

5. High-level Description of the Proposed Cryptographic Schemes

Based on the platform features and functionalities, the following scenario depicted in Figure 1 is considered. At the top of the hierarchy is the main cloud, which comprises power stations, distribution services, and management services and operations (e.g., SCADA services integrated into the cloud). There are regional "*core*" network nodes under the cloud layer, and "*neighborhood*" nodes that both constitute part of the secured pathway to the cloud. The fog nodes come under the neighborhood nodes, and are intermediary arranged between IoT devices and cloud servers taking cloud computing to the edge of the network. The security toolbox is part of the Eucalyptus dashboard, which is hosted between the thing layer and fog layer to facilitate the encryption/decryption of services offered at the fog layer. These cryptographic services imply that all requests from IoT devices must go through it first to authenticate, encrypt and assigns keys accordingly. This will help to eliminate the requirement for checking the validity of sending/receiving certificates, which slows down the communication process. Two well-established cryptographic schemes will be available, namely "*identity-based encryption*", and "*identity-based signature*". Moreover,

the fog nodes provide several general user services and computing resources, managing and controlling connected devices in the region and process information received from the IoT devices. The IoT generated data that is encrypted by the cryptographic services at fog layer is forwarded to the cloud via "neighborhood" nodes and regional "core" network nodes.

In traditional scenarios, the identity-based encryption scheme requires a "*trusted*" external party, dubbed a Private Key Generator (PKG) who generates a secret master key (*mk*) and a public parameter (*params*) from selected security parameters denoted in our model by (Ω). In our proposed platform, the PKG will be located in the fog layer as part of the proposed End-to-End Security Toolbox service for three main reasons:

a. In traditional systems, the PKG is identified as the main contributor in slowing down the communication process. Because of network delays, delivering the master key and associated parameters to other trusted parties is time consuming. Hence, there is a need to keep the PKG as close as possible to those parties. To this end, the usage of the fog layer becomes an appealing solution.

b. As mentioned above, the PKG needs to be a trusted third party, in which case there might be a long path (i.e., multiple hops) between the PKG and the user(s), which increases the security concerns and risks. In contrast, if the PKG is located in the fog layer, it will take a single hop from the sender and a secured channel can then be established between the PKG and the users to prevent eavesdropping.

c. One of the key issues in deploying security solutions based on public key cryptography is the high cost of maintaining the involved key infrastructure. As such, we argue that adopting identity-based cryptography and hosting the PKG in the fog layer offer a cost-effective security solution.

As illustrated in Figure 2, the *params* will be given to every beneficiary in the system including the sender and receiver. First the receiver authenticates him/herself to the PKG by submitting his/her identity, denoted by ID_{rec}, which can be any string such as an email address, a telephone number, etc. Next, the PKG computes the private key $K_{ID_{rec}}$ associated with ID_{rec} identities by running the private key extraction algorithm (*Extract* in Figure 2) and by providing its master secret key *mk* and *params* as input.

In this scenario, any sender who possesses ID_{rec} and *params*, can encrypt a plain message *M* into a cipher text *C* using the *Encrypt* algorithm. Once *C* is delivered, the receiver decrypts it using *Decrypt* algorithm with the private key $K_{ID_{rec}}$ (obtained previously from the PKG) as input. Figure 2 depicts the essential operations of the proposed identity-based encryption scheme.

Similarly, as illustrated in Figure 3, when the signer submits his/her identity *IDsig*, the PKG computes the private key $K_{ID_{sig}}$ associated with ID_{sig} using the (*Extract*) with the master secret key *mk* and a public parameter (*params*) from selected entered security parameters denoted in our model by (Ω). Using $K_{ID_{sig}}$, the signer can obviously sign a message *M* to create a signature *s* using the (*Sign*) algorithm. Providing the message *M*, the signer's identity ID_{sig}, and the signature s, any party (i.e., verifier) can validate the signature *s* by running *Verify*.

In summary, our approach ensures that a user, with a registered identifier and who possesses the corresponding private key will be the only one who can decrypt or produce a valid signature, even if the fog node is accessible by other users. This approach ensures authentication of the user and protects the message from modification thus helping to maintain data integrity while in transit.

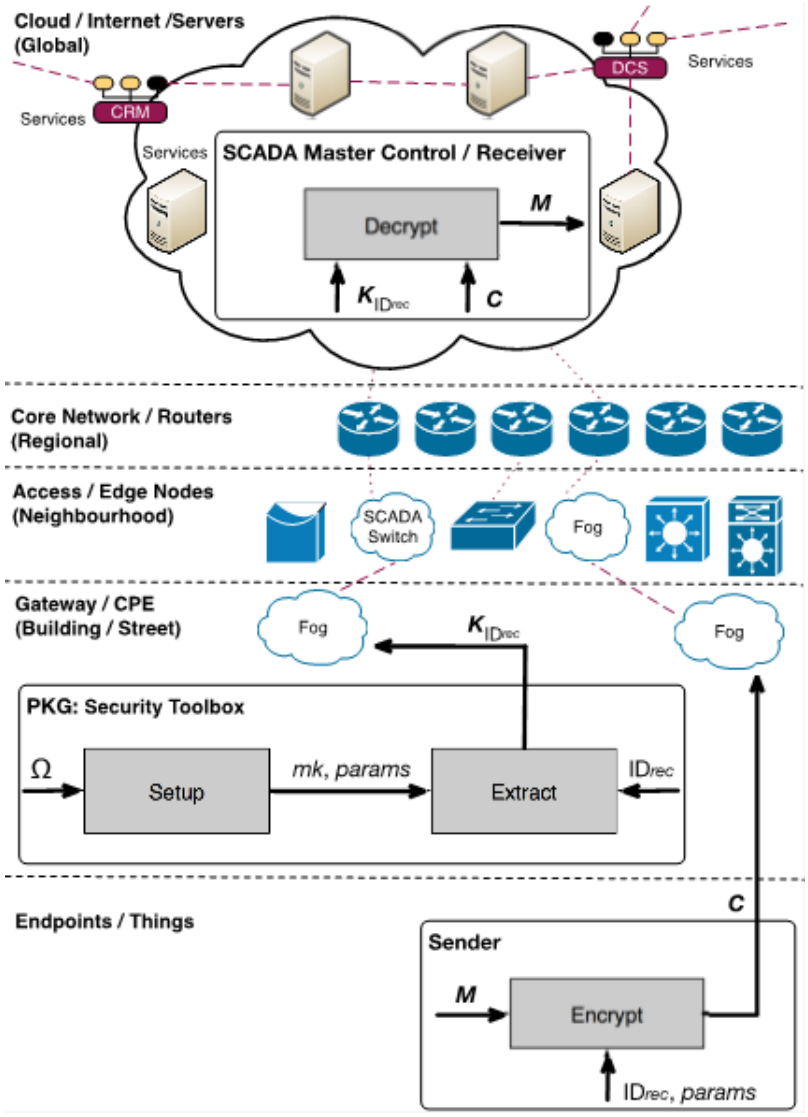

Figure 2: Identity-based encryption on the fog

6. Prototype Implementation

6.1. Experimental Setup

A prototype has been built based on Eucalyptus[1] open-source solution, which provides the necessary infrastructure components as a service to the users of the platform. On top of Eucalyptus, we provide the proposed platform services, as shown in Figure 1. We used Java as a programming language and NetBeans 8.2 with Payara Server 5.181 as the Integrated Development Environment (IDE) in order to implement the proposed prototype. The experiments were run on an Apple iMac (Retina 5K display, 4.2 GHz Intel Core i7, and 46 GB 2400 MHz DDR4). In addition, the encryption/decryption services of the system were built using *JPair*[2], which allows bilinear pairing with the cloud via Eucalyptus. The encryption/decryption services are hosted on the fog layer, which implies that all requests must go through it first in order to authenticate, encrypt and assigns keys accordingly.

Eucalyptus provides a control panel (i.e., dashboard) for remotely hosted resources. This light-weight dashboard application is installed on the user's machine. Via this dashboard, the user can

[1] Is an open-source cloud software for creating AWS based private, public and hybrid cloud computing environment and services

[2] Java-based cryptographic library available online at http://homepages.cs.ncl.ac.uk/changyu.dong/jpair/intro.html

specify the resource(s) needed (including what, when, where, and by whom). Therefore, a full description of the requested resource(s) on the cloud end will be provided. Then, the system generates a resource *"workflow"* to launch the corresponding resources with the proper set of parameter values at the time specified. In the subsequent sections, we present the implementation details of a prototype of our Secure Fog-based Platform (SeFoP) and we will validate it through simulations.

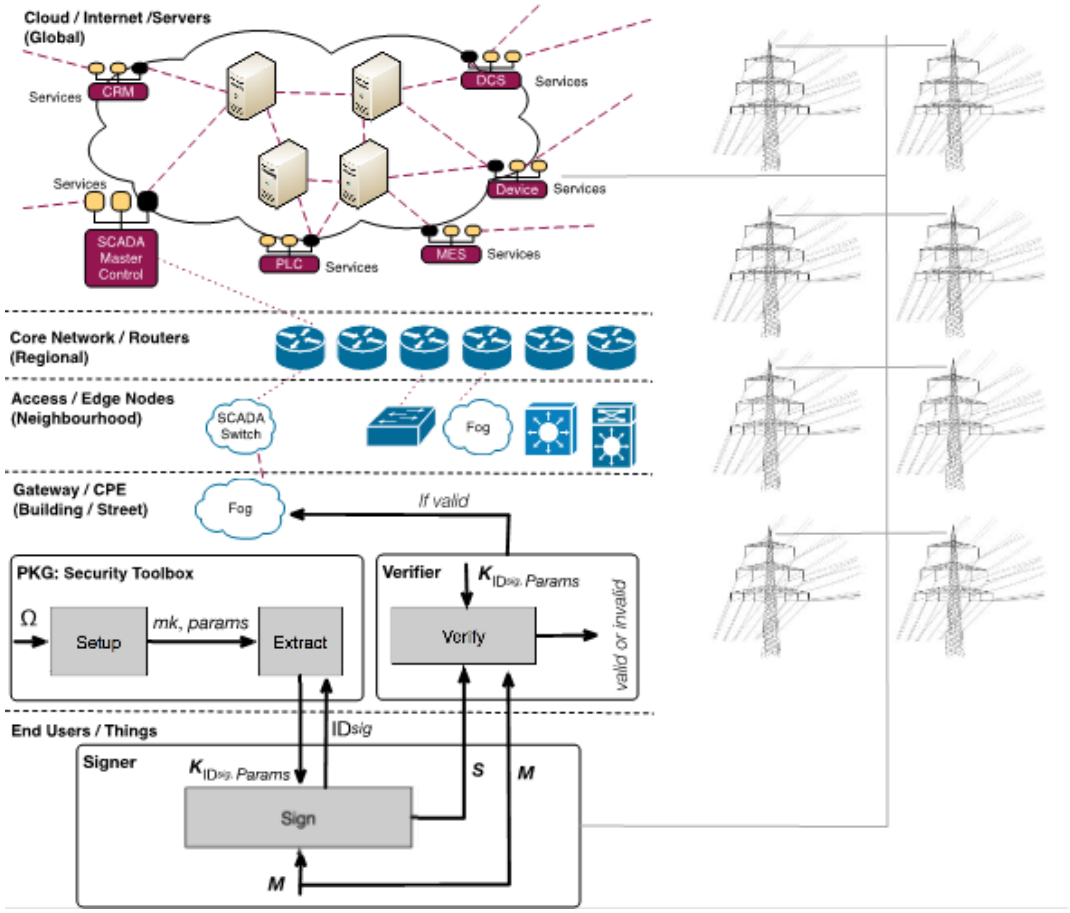

Figure 3: Identity-based signature on the fog

6.2. System Design and Implementation

The proposed SeFoP has been implemented based on the system architecture depicted in Figures 1, 2 and 3. The following features/services were implemented in SeFoP on top of Eucalyptus:

a. *Add/delete identity*: This enables all system components that are involved in sending and receiving data (including senders and receivers and other resources in the system), to add/register their IDs before they will be able to use/communicate-with other system components. This can be done via *addResource* method as illustrated in Figure 4. Once a new component's ID has been registered, the private key for that object will be created in the form of *KIDsig* as shown in Figure 3. In the same vein, if the component is no longer available, its ID can be removed from the system using the *deleteResource* method in which case that the particular component will not be able to send/receive messages (or use the system) thereafter.

b. *Request encryption/decryption*: Once a receiver registers its identity, the created ID will be available as a (*Public key*), which can be used by any sender (as a Private key) to direct the message to that particular entity. Accordingly, data will be encrypted using the private key

of the receiver before being sent. This ensures that even if the data is interrupted, it will not be readable by the interceptors. This is made possible by using the receiver's registered ID as a private key to encrypt the message. Once the receiver who holds the private key receives the message, he is the only entity that can decipher the message using its private key.

These services are designed to be integrated as part of the fog-based user's application so that the user can seamlessly send requests to access designated SCADA nodes or services in the cloud. Algorithm 1 details how the encryption and decryption requests will be performed in SeFoP. (Procedure 1 – Request Encryption) starts in lines 2 and 3 with initiating a user requests (u_r) and a Virtual Machine (v_m). The request will then be sent to a fog node in F_n where $F_n = \{f_1, f_2, f_3, .., f_n\}$ for encryption as per line 4. The *Encrypt* function (lines 5-12) encrypts the incoming request using the IP address of the desired server (s_{ip}). Using s_{ip} helps in implementing 3-levels encryption: (1) adding more elements (i.e., bytes) to the request via inserting the IP of the server (*addBytes*, line 8), (2) shifting the elements (*shiftRows*, line 9) by the size of $s_{ip;}$ and, (3) mixing the elements of the massage via (*mixColumns*, line 10). (Procedure 2 – Request Decryption) works in a reverse-wise, as shown in (lines 15-24). The decryption is achieved by removing and reversing the elements of the request message, thus only the server with the correct IP address reads the message correctly.

Algorithm 1: ENCRYPTION AND DECRYPTION

Input: UserRequest (u_r); VirtualMachine (v_m); ServerIP (s_{ip})
Parameters : FogNode (f_n); EncryptedMsg(e_m)
Initialisation: $u_r = \phi$; $v_m = \phi$

1 **Procedure 1.** *Request Encryption* **by**
2 $u_r = getUserRequest()$; ▷ initiate u_r
3 $v_m = getVirtualMachine()$; ▷ initiate v_m
4 $F_n \leftarrow$ sendToFog(u_r, v_m) ;
5 **Function** *Encrypt*(u_r, v_m, s_{ip}) **by**
6 $e_m = initiateMsg(u_r, v_m, s_{ip})$;
7 **for** *each element* $\in e_m$ **do**
8 $e_m = addBytes(e_m)$;
9 $e_m = shiftRows(e_m)$;
10 $e_m = mixColumns(e_m)$;
11 **end**
12 **End**
13 return e_m ;
14 **End**
15 **Procedure 2.** *Request Decryption* **by**
16 get(e_m) ;
17 **Function** *Decrypt*(e_m) **by**
18 $e_m = initiateMsg(e_m, s_{ip})$;
19 **for** *each element* $\in e_m$ **do**
20 $e_m = subBytes(e_m)$;
21 $e_m = reverseRows(e_m)$;
22 **end**
23 **End**
24 return e_m ;
25 **End**

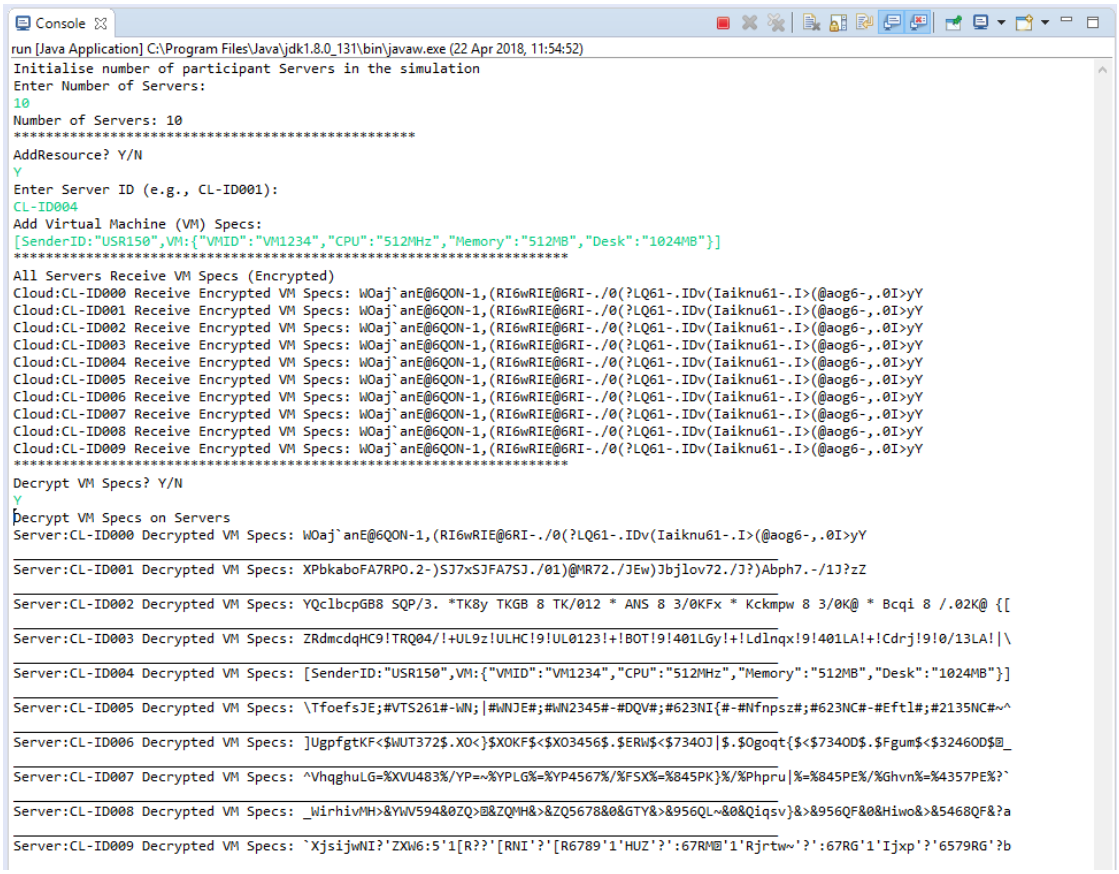

Figure 4: SeFoP Service Implementation

To evaluate the performance of the proposed SeFoP at runtime, it is important to benchmark the results against well-established and published models. Therefore, the Multi-level User Access Control Layer (MLAC) [47] scheme for adding additional security/authentication layer on the cloud was considered as a benchmark for the purpose of performance comparison. It should be noted that we use identical simulation parameters of MLAC in order to achieve a systematic and consistent performance comparison. Hence, we consider a specific scenario of adding and removing VMs and storage on a specific server on the cloud and then we monitor and compare the response time of adding and removing these resources under three scenarios: the proposed SeFoP approach, the MLAC approach [47], and the no security approach.

Suppose that a user with ID "USR150" sends a request to create a VM with certain specifications on server hosted on a public cloud and identified by a string "CL-ID004". The user's request in this system will be "VMID:VM1234, CPU:512MHs, Memory:512MB, Desk:1024MB". As such, the scenario works in the following sequence:

- First, all servers on the cloud will register their identity using the *Add identity* service. Afterwards, a private key will be generated for each server on the fog layer, using the ID of that server.
- The user sends the request to a particular server via the fog layer, using the *request encryption* service.
- Finally, the server uses its private key to decipher the cipher request via *decrypt request*.

7. Discussion and Evaluation Results

Although a fully completed evaluation of the proposed platform is underway, some preliminary results for the execution of the proposed "*SeFoP*" features will be presented in this section. The following testing was conducted primarily to monitor the response-time of adding and removing

VM(s) with SeFoP and to compare the results against those obtained with the MLAC approach [47], and those obtained with no implemented security features. Evidently, introducing a new behavior (SeFoP in this case) to the service has a direct performance impact due to the additional overhead associated with the additional code interpretation needed to execute and link the new service. However, given that SeFoP is presented as part of the fog main functions (i.e., embedded as part of the connection network), it does not have any negative impact on the system performance; but showed performance improvement instead. Hence, it is suitable for real-time critical systems in which the response time and latency are key factors. In fact, as may be seen in Figure 5, when adding or removing cloud resources (VMs and storage in our case), the overall system performance improves with SeFoP compared to MLAC, and this performance is remarkably comparable to a that of a plain-text solution approach (i.e., with no security features). This is because the new added SeFoP framework is available on the route between the user and the cloud data center (i.e., fog layer); it is not outsourced to a third party. In Figure 5, the dark grey bars represent the response time (in *msec*) of the application using SeFoP, while the light bar in the middle represents the application response-time with MLAC.

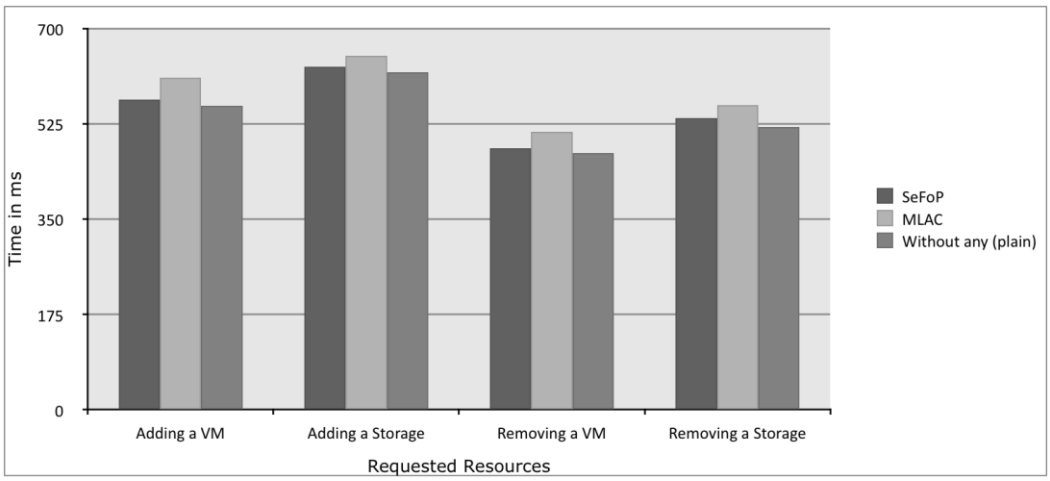

Figure 5: Overheads of Interacting with the Application: SeFoP vs MLAC vs no Security

A further system performance testing has been conducted to compare the stability of the proposed SeFoP framework (when adding and removing resources from/to the cloud) against the corresponding results obtained with the MLAC and with the "no security" approaches. The results shown in Figure 6 showcase the relative robust stability of SeFoP when compared to the other two approaches.

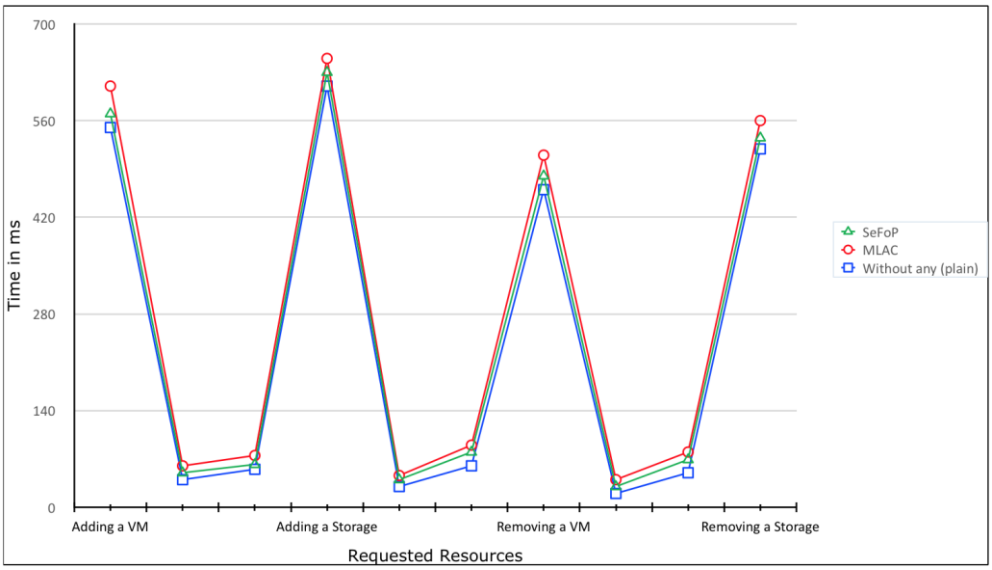

Figure 6: System stability Comparison

Finally, we compared the scalability of the system with and without SeFoP features. For this purpose, we, first, ran multiple simulations that involve an increasing number of VMs (5 – 50) and measured the overall performance with and without SeFoP. Our results, as shown in Figure 7, illustrate that compared to MLAC, the performance gain brought by SeFoP is around 2.8 seconds when adding 5 VMs and 3.2 seconds when adding 10 VMs; and it is less than 7 seconds in other instances. Then, we ran another experiment to evaluate performance of SeFoP against MLAC when adding an extreme number of VMs (1000 VMs) all at the same time. Although, the results/bars in Figure 8 show time difference among the three used approaches; however, this difference is a fraction of milliseconds and it is not noticeable by the end users. The total time needed to add 1000 VMs through SeFoP is around 112 seconds. As such, we are satisfied that, at this stage, our approach will not introduce any overhead over larger jobs on the cloud.

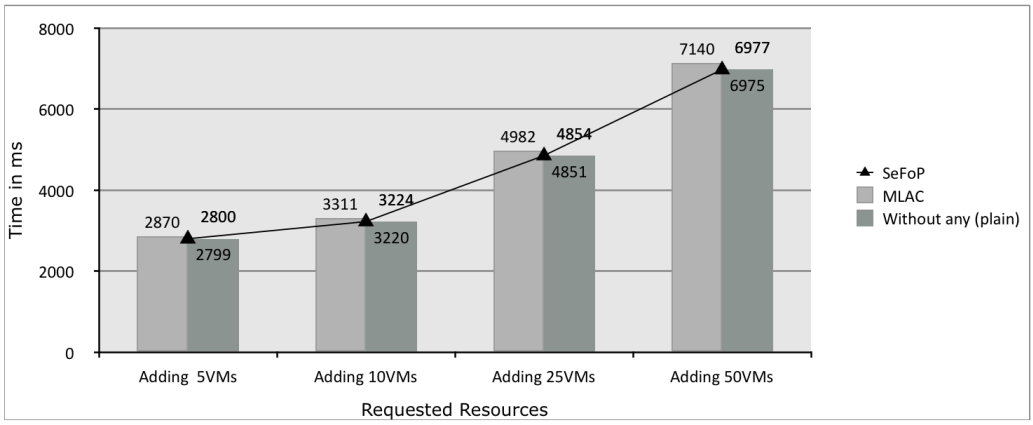

Figure 7: System Scalability Comparison

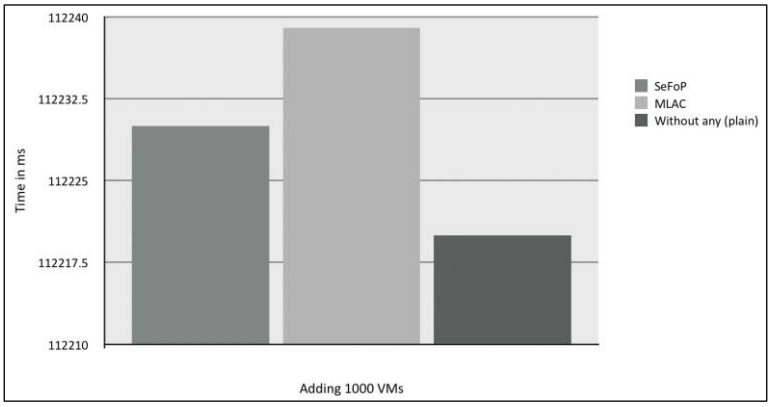

Figure 8: System Scalability when Adding 1000 VMs

8. Conclusions

There is a clear need today for more proven secure platforms to facilitate the adoption and diffusion of IoT-based services among CI providers. This requires effective mechanisms to secure the distributed cloud services, the end-to-end interconnected network, and the overall fog infrastructure. These issues are now starting to be tackled based on existing enterprise-wide technologies on the basis that they can provide a reasonable level of assurance over the security of the platform while limiting the cost and complexity of the overall approach.

In this paper, we have demonstrated how cloud computing and the fog layer can practically be made secure to support CI providers. In addition, we have proposed a dynamic and open platform to secure CI in hybrid cloud/fog environments, and we identified the main elements of a 'security toolbox' that providers can develop and deploy to orchestrate this process. Finally, to our knowledge, this is the first contribution that aims to explore the usage of identity-based cryptography and identity-based signature schemes within the fog layer, which was also substantiated by a performance evaluation simulation. Our future work will focus on validating the proposed approach against CI provider expectations and on further developing, integrating and evaluating the identified functionalities to comprehensively demonstrate the advantages of the proposed toolbox.

Acknowledgments: This research is partially supported by Research Incentive Funds (R15046) and Cluster Research Award (R16083), Zayed University, Abu Dhabi, United Arab Emirates. This grant does not cover costs to publish in open access.

References

[1]. Clinton Dsouza, Gail-Joon Ahn, and Marthony Taguinod. "Policy-driven security management for fog computing: Preliminary framework and a case study." In Information Reuse and Integration (IRI), 2014 IEEE 15th International Conference on, pp. 16-23. IEEE, 2014.

[2]. Shanhe Yi, Zijiang Hao, Zhengrui Qin, and Qun Li. "Fog computing: Platform and applications." In Hot Topics in Web Systems and Technologies (HotWeb), 2015 Third IEEE Workshop on, pp. 73-78. IEEE, 2015.

[3]. Redowan Mahmud, and Rajkumar Buyya. "Fog Computing: A Taxonomy, Survey and Future Directions." arXiv preprint arXiv:1611.05539 (2016).

[4]. Muhammad Tanvir Alam. "A Review on Cloud-based Privacy Preserving Schemes for Smart Meters." INFOCOMP Journal of Computer Science 15, no. 1 (2016): 19-33.

[5]. PYL, T.V.D.: Monitoring and control. White chlorine-free paper, European Commission: Information Society and Media (November 2008).

[6]. HyungJun Kim. "Security and vulnerability of SCADA systems over IP-based wireless sensor networks." International Journal of Distributed Sensor Networks (2012).

[7]. Pierluigi Paganini. (2013). SCADA and Security of Critical Infrastructures. http://resources.infosecinstitute.com/scada-security-of-critical-infrastructures/ (accessed 3 May 2017).

[8]. CPNI: Centre for the Protection of National Infrastructure. http://www.cpni.gov.uk (accessed 5 April 2017).
[9]. Martin Rudner. "Cyber-threats to critical national infrastructure: An intelligence challenge." International Journal of Intelligence and Counter Intelligence 26, no. 3 (2013): 453-481.
[10]. Dorothy Denning. "Stuxnet: What has changed?." Future Internet 4, no. 3 (2012): 672-687.
[11]. Hitomi Takagi, Takahito Morita, Masafumi Matta, Hiroki Moritani, Takashi Hamaguchi, Sun Jing, Ichiro Koshijima, and Yoshihiro Hashimoto. "Strategic security protection for industrial control systems." In Society of Instrument and Control Engineers of Japan (SICE), 2015 54th Annual Conference of the, pp. 986-992. IEEE, 2015.
[12]. PROTECT: Research Centre for Critical Infrastructure Computer Technology and Protection. http://www.protect-ci.org (accessed 8 May 2018).
[13]. Stouffer, KS Keith, and Joe Falco. "Recommended practise: Improving industrial control systems cybersecurity with defense-in-depth strategies." Department of Homeland Security, Control systems security program, national cyber security division (2009).
[14]. Michael Mackay, Adil Al-Yasiri. Thar Baker. "Security-oriented cloud computing platform for critical infrastructures." Computer Law and Security Review 28(6), 679-686 (2012).
[15]. Thar Baker, Muhammad Asim, Hissam Tawfik, Bandar Aldawsari, and Rajkumar Buyya. "An Energy-aware Service Composition Algorithm for Multiple Cloud-based IoT Applications." Journal of Network and Computer Applications (2017).
[16]. Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. "Security and privacy challenges in industrial internet of things." In Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE, pp. 1-6. IEEE, 2015.
[17]. Feyza Yildirim Okay, and Suat Ozdemir. "A fog computing based smart grid model." In Networks, Computers and Communications (ISNCC), 2016 International Symposium on, pp. 1-6. IEEE, 2016.
[18]. Berthold Bitzer, and Enyew Sileshi Gebretsadik. "Cloud computing framework for smart grid applications." In Power Engineering Conference (UPEC), 48th International Universities, pp. 1-5. IEEE, 2013.
[19]. Jinsung Byun, Youngil Kim, Zion Hwang, and Sehyun Park. "An intelligent cloud-based energy management system using machine to machine communications in future energy environments." In Consumer Electronics (ICCE), 2012 IEEE International Conference on, pp. 664-665. IEEE, 2012.
[20]. Marcelo Yannuzzi, R. Milito, René Serral-Gracià, Darwin Montero, and Mario Nemirovsky. "Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing." In Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014 IEEE 19th International Workshop on, pp. 325-329. IEEE, 2014.
[21]. Shanhe Yi, Zhengrui Qin, and Qun Li. "Security and privacy issues of fog computing: A survey." In International Conference on Wireless Algorithms, Systems, and Applications, pp. 685-695. Springer International Publishing, 2015.
[22]. Ivan Stojmenovic, Sheng Wen, Xinyi Huang, and Hao Luan. "An overview of Fog computing and its security issues." Concurrency and Computation: Practice and Experience 28, no. 10 (2016): 2991-3005.
[23]. Federal Trade Commission. "Internet of Things: Privacy & security in a connected world." Washington, DC: Federal Trade Commission (2015).
[24]. Nechibvute, Action, and Courage Mudzingwa. "Wireless sensor networks for scada and industrial control systems." International Journal of Engineering and Technology 3, no. 12: 1025-1035, (2013).
[25]. Ltd, W.L.C.A.C.P.: Autonomous Remote SCADA. http://www.wideye.com.sg/default/index.php/remote-scada [accessed 8 May 2017]
[26]. Kanghyo Lee, Donghyun Kim, Dongsoo Ha, Ubaidullah Rajput, and Heekuck Oh. "On security and privacy issues of fog computing supported Internet of Things environment." In Network of the Future (NOF), 2015 6th International Conference on the, pp. 1-3. IEEE, 2015.
[27]. Anam Sajid, Haider Abbas, and Kashif Saleem. "Cloud-assisted IoT-based SCADA systems security: A review of the state of the art and future challenges." IEEE Access 4 (2016): 1375-1384.
[28]. Flavio BonomiBonami, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. "Fog computing: A platform for internet of things and analytics." In Big Data and Internet of Things: A Roadmap for Smart Environments, pp. 169-186. Springer International Publishing, 2014.

[29]. Dastjerdi, Amir Vahid, Harshit Gupta, Rodrigo N. Calheiros, Soumya K. Ghosh, and Rajkumar Buyya. "Fog computing: Principles, architectures, and applications." arXiv preprint arXiv:1601.02752 (2016).
[30]. Luis M. Vaquero, and Luis Rodero-Merino. "Finding your way in the fog: Towards a comprehensive definition of fog computing." ACM SIGCOMM Computer Communication Review 44, no. 5 (2014): 27-32
[31]. Subhadeep Sarkar, Subarna Chatterjee, and Sudip Misra. "Assessment of the Suitability of Fog Computing in the Context of Internet of Things." IEEE Transactions on Cloud Computing, no. 99, pp. 1, (2015).
[32]. Kirak Hong, David Lillethun, Umakishore Ramachandran, Beate Ottenwälder, and Boris Koldehofe. "Mobile fog: A programming model for large-scale applications on the internet of things." In Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing, pp. 15-20. ACM, 2013.
[33]. Cuong T. Do, Nguyen H. Tran, Chuan Pham, Md Golam Rabiul Alam, Jae Hyeok Son, and Choong Seon Hong. "A proximal algorithm for joint resource allocation and minimizing carbon footprint in geo-distributed fog computing." In Information Networking (ICOIN), 2015 International Conference on, pp. 324-329. IEEE, 2015.
[34]. Mohammad Aazam, and Eui-Nam Huh. "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT." In Advanced Information Networking and Applications (AINA), IEEE 29th International Conference on, pp. 687-694. IEEE, 2015.
[35]. Harishchandra Dubey, Jing Yang, Nick Constant, Amir Mohammad Amiri, Qing Yang, and Kunal Makodiya. "Fog data: Enhancing telehealth big data through fog computing." In Proceedings of the ASE BigData & SocialInformatics, p. 14. ACM, 2015.
[36]. John K. Zao, Tchin Tze Gan, Chun Kai You, Sergio José Rodríguez Méndez, Cheng En Chung, Yu Te Wang, Tim Mullen, and Tzyy Ping Jung. "Augmented brain computer interaction based on fog computing and linked data." In 2014 International Conference on Intelligent Environments (IE), pp. 374-377. IEEE, 2014.
[37]. Á. MacDermott, Q. Shi, M. Merabti, and K. Kifayat, "Protecting Critical Infrastructure Services in the Cloud Environment," in Proceedings of the 12th European Conference on Information Warfare and Security (ECIW), 2013, pp. 336–343.
[38]. Richard Piggin. "Securing SCADA in the cloud: Managing the risks to avoid the perfect storm." In Instrumentation Symposium 2014, IET & ISA 60th International, pp. 1-6. IET, 2014.
[39]. Nuno Santos, Krishna P. Gummadi, and Rodrigo Rodrigues. "Towards trusted cloud computing." In: HotCloud'09 Proceedings of the 2009 Conference on Hot Topics in Cloud Computing. ACM Digital Library, 9(9), p.3, 2009.
[40]. Flavio Lombardia, Roberto Di Pietro. "Secure virtualization for cloud computing." Journal of Network and Computer Applications 34(4), 1113-1122 (2011)
[41]. Philippe Gourbesville, Jelena Batica, Jean Yves Tigli, Stephane Lavirotte, Gaetan Rey, and Durairaju Kumaran Raju. "Flood warning systems and ubiquitous computing." La Houille Blanche 6 (2012): 11-16.
[42]. Omner Barajas, How the Internet of Things (IoT) Is Changing the Cybersecurity Landscape. https://securityintelligence.com/how-the-internet-of-things-iot-is-changing-the-cybersecurity-landscape/, online 04, (2017).
[43]. Arwa Alrawais, Abdulrahman Alhothaily, Chunqiang Hu and Xiuzhen Cheng, "Fog Computing and the Internet of Things: Security and privacy issues", IEEE Internet Computing, 21, no.2: 31-42, (2017).
[44]. Eduardo B. Fernandez, Nobukazu Yoshioka, Hironori Washizaki, and Madiha H. Syed. "Modeling and Security in Cloud Ecosystems." Future Internet 8, no. 2: 13, (2016).
[45]. Ivan Stojmenovic, and Sheng Wen. "The fog computing paradigm: Scenarios and security issues." In Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on, pp. 1-8. IEEE, 2014.
[46]. Áine MacDermott, Qi Shi, Madjid Merabti, and Kashif Kifayat, "Hosting critical infrastructure services in the cloud environment considerations" in InderScience International Journal of Critical Infrastructures, 12/2014; 10(3).

[47]. Thar Baker, Michael Mackay, Amjad Shaheed, Bandar Aldawsari, “Security-Oriented Cloud Platform for SOA-Based SCADA”, 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), 2015, pp. 961-970.