sensors

MDPI

Article

Hardware-Intrinsic Multi-Layer Security: A New Frontier for 5G Enabled IIoT

Hussain Al-Aqrabi [1]*, Anju P. Johnson [2], Richard Hill [1], Phil Lane [1], and Tariq Alsboui [1]

[1] Department of Computer Science, Centre for Industrial Analytics (CIndA), School of Computing and Engineering, University of Huddersfield, Queensgate, Huddersfield HD1 3DH, UK
[2] Department of Engineering and Technology, Centre for Planning, Autonomy and Representation of Knowledge (PARK), School of Computing and Engineering, University of Huddersfield, Queensgate, Huddersfield HD1 3DH, UK
* Correspondence: H.Al-Aqrabi@hud.ac.uk

Received: 27 February 2020; Accepted: 28 March 2020; Published: 31 March 2020

Abstract: The introduction of 5G communication capabilities presents additional challenges for the development of products and services that can fully exploit the opportunities offered by high bandwidth, low latency networking. This is particularly relevant to an emerging interest in the Industrial Internet of Things (IIoT), which is a foundation stone of recent technological revolutions such as Digital Manufacturing. A crucial aspect of this is to securely authenticate complex transactions between IIoT devices, whilst marshalling adversarial requests for system authorisation, without the need for a centralised authentication mechanism which cannot scale to the size needed. In this article we combine Physically Unclonable Function (PUF) hardware (using Field Programmable Gate Arrays—FPGAs), together with a multi-layer approach to cloud computing from the National Institute of Standards and Technology (NIST). Through this, we demonstrate an approach to facilitate the development of improved multi-layer authentication mechanisms. We extend prior work to utilise hardware security primitives for adversarial trojan detection, which is inspired by a biological approach to parameter analysis. This approach is an effective demonstration of attack prevention, both from internal and external adversaries. The security is further hardened through observation of the device parameters of connected IIoT equipment. We demonstrate that the proposed architecture can service a significantly high load of device authentication requests using a multi-layer architecture in an arbitrarily acceptable time of less than 1 second.

Keywords: Internet of Things; cloud computing; hardware security; field programmable gate array (FPGA); 5G; analytics

1. Introduction

Adopting evolving business models that are enabled by emerging 5G technologies is a challenge when attempting to maintain legitimate security and privacy considerations for Internet of Things (IoT) and Industrial Internet of Things (IIoT) devices [1]. It is clear that raising industrial users' knowledge that a substantial amount of the interest they create is intrinsically connected with intellectual property (IP) ownership and continuous development. There is also the persistent risk of a security breach that could compromise ownership of the IP, putting the underlying business model at higher risk [2,3]. This is particularly prevalent in the provision of IoT assisted healthcare systems, which is a pertinent example of distributed IT systems that have similarly complex needs and stakeholder requirements[4]. Although cloud computing illustrates how technologies and business models are used to provide new business opportunities to enterprises, businesses remain at risk of emerging threats due to the proliferation of cloud services, including multi-tenant cloud environments [5,6]. The promise of 5G

Infrastructure holds immense possibilities for greater integration of physical devices that are ideally suited to IIoT for several reasons, as follows:

- *Less network latency* increases overall response times and is able to enhance security protocol strictness without sacrificing the system's user experience;
- *Higher data rates* enable the sharing of data between devices, and the utilisation of metadata to support secure transactions building trust between devices;
- *Lower power demand* allows widespread use of sensing and processing devices where power infrastructure is absent.

The huge advantage of millimetre wave (MMW) radio spectrum for 5G is a crucial enabler for better network performance, although at a loss of propagation range [2]. Whereas the higher frequency band has specific physical security [7,8], this approach is not one that we should depending on. A manipulative attacker seated beside the IIoT device may be able to transmit data externally [9,10] [11,12]. The heterogeneous nature of IoT communications with its heterogeneous architecture and devices, requires information sharing and collaboration across a wide range of networks. This poses severe privacy and security issues [13]. IoT privacy protection seems to be more vulnerable than conventional Information and Communication Technology (ICT) systems because of several vector threats against IIoT technologies [14,15]. Modelling these vulnerabilities is challenging, particularly since the multiplicity of IIoT devices each represent agents within a complex system of interactions that need to be secure[16–18].

Consequently, there is a need to create a flexible multi-layer cloud security architecture that provides adequate authentication for multiple parties in a reliable way, while being mindful of the heterogeneous nature of how IIoT devices will communicate efficiently. Our article discusses how well the cloud methodology was developed to guide the creation of the security architecture for several purposes. Firstly, cloud computing architectures actively support complex demands via elasticity [19,20] and facilitates the standardisation of diverse systems by abstraction. Secondly, there seems to be a proven architectural reference model given by NIST [21], which is widely used. Lastly, cloud systems have similar features with IIoT systems in that multiple parties need to function together and collaborate by a secure exchange of data and assets [5].

Previous work addressed the specific instance of multi-party trust authentication for the deployment of cloud based business intelligence systems. The authors also have built and adapted to accommodate a particular instance where the introduction of 5G network services would enable new business opportunities through increased efficiency. To support these features, the authors extended cloud-based infrastructure to include Physically Unclonable Function (PUF) hardware. Since the PUFs are resilient to spoofing attacks, the PUF hardware offers a higher level of security toward direct physical attacks, which are essential in situations where there is a need to rapidly authenticate several parties to ensure trustworthy connections [5].

The delivery of analytical resources from manufacturing plant represents a real scenario that the authors addressed, allowing the secure exchange of heterogeneous data, and also performance appraisal, between both the IIoT components and the enterprise (ICT) system of the organisation, often using Micro Services architecture[22]. This article considers the potential adversarial attacks to consider on such a device, which assists the design of an agile approach to multi-layer security. The authors created algorithms that require authentication through PUFs to provide effective, secure, and flexible access to IoT cloud applications. The article is arranged as follows. Section 2 defines a framework for multi-layer security. Section 3 presents a related secure solution for networking which utilises PUFs. In Section 4, we present the results of experiments that illustrate the potential for this approach. Finally, we conclude in Section 5.

2. Multi-Layer Security Model

The critical challenge for IIoT is the implementation and processing the large amount of data produced by these devices. In attaining this IoT vision, Low-Power (LP) and Loss Networks (LLNs) are

diversified, and the interconnection of restricted physical devices by the use of LP and LLNs involves the modification of protocols and existing structures currently in common use [23]. Latterly, hardware trojan attacks have developed as a threat to all hardware and integrated circuits (ICs) [24]. The main challenge of handling network connectivity in a tightly-equipped setting, including a smart factory, is to identify and manipulate different attack vectors. In principle, the promise of cloud resources also introduces potential system vulnerabilities. As such, the authors opted to create a security model that divides a variety of security controls through multiple layers of defences [25]. Figure 1 illustrates a proposed secure architecture. The authors use the example of a traditional enterprise infrastructure with analytics capabilities to promote tactical and organisational business decision-making.

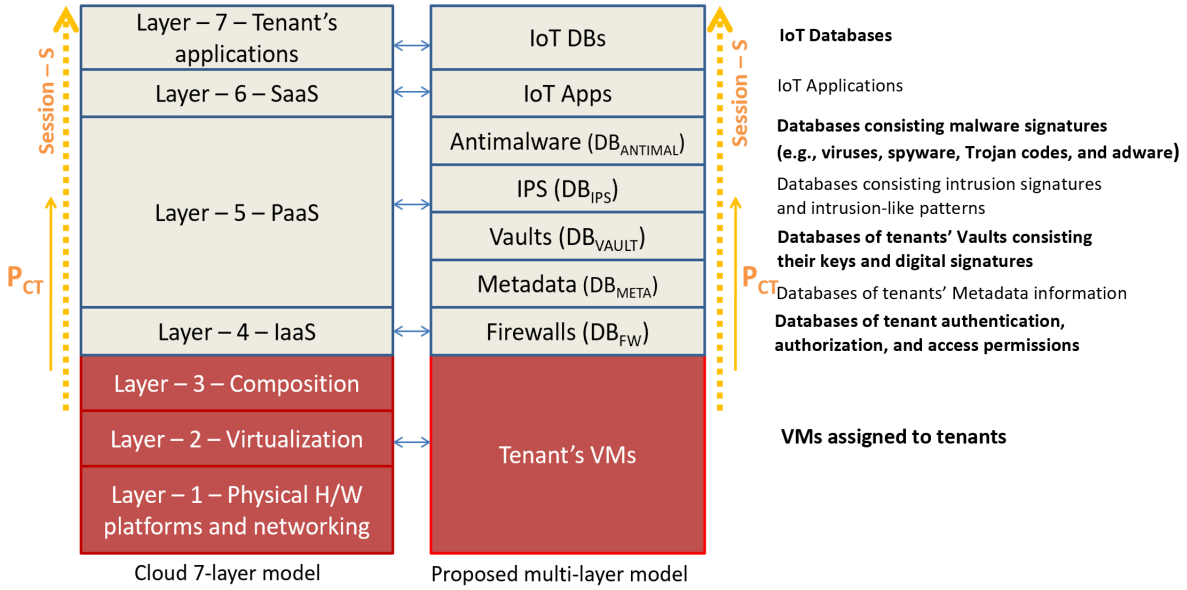

Figure 1. Multi-layer security proposed model [2].

Primarily, as just that, our model was examined, in which individual users are tenants in a multi-tenant cloud environment. In our model, we consider the case that each user or (IIoT device or sensor) is described by a multi-cloud enterprise system as a prospective tenant. As the architecture enables the abstraction of resources, users that also require access to the business network can do this remotely, through virtual machines, and also through hardware devices [25].

All endpoints are secured via firewalls. In the beginning, all external requests are assisted by authentication data firewalls for each potential tenant. The Metadata layer, for example offers security controls for the features previously allowed for each tenant registered. The lack of required authentication data will prevent the user from effectively communicating with the system. Once the simple authentication is established, a Tenant Metadata Layer maintaining rules-based controls is required to determine which part of the business system a permitted tenant can access. For instance, this may apply to specific databases or reports. While the IIoT device offers data for a variety of analytics processing, which involves not only adding data to the repository as well as maintaining access to other data sources which can be collected and merged to present better analytical services.

A secure connection must be established, and this has been achieved by using the public-key Infrastructure (PKI). The PKI uses it to verify that the signature is authentic. Within its model layer, public key certificates are preserved within the Digital Vault, and this offers another secure degree where the user session may be approved or removed. In the case where the deceptive attackers penetrated aggressively into the first three layers, Layer four offers a deeper layer of protection. Whereas the controls of the prior layers are capable of protecting against various attacks, these can not prevent them from a harmful intruder who previously has the authority to access the system. The

network will monitor suspicious activities using the Intrusion Prevention System as well as to detect irregular actions, in order to set up a session for tenants engaging in inappropriate behaviour.

An anti-malware layer of protection reinforces layer four. Far more surreptitious activity, for example, hidden executable code, may disrupt as it is implemented into the business network. Layer 5 keeps an activity record, and a list of known threats. Within the application cloud layer, this layer comprises the business features and is of considerable value to enterprise clients. During that time, the client has entered this layer, simple authentication, client verification by PKI, intrusion prevention system (IPS) and anti-malware inspections were already made, with each layer being able to terminate the session. Apart from business applications, it is necessary to access corporate repositories by a particular type of user, whether directly through application programming interfaces (API) or through querying and monitoring interfaces, usually provided via a web portal [26].

3. NIST Cloud Model

NIST is developing standard protocols and guidelines for user or client devices access to the Cloud by means of an interface for virtualisation, Internet browser interface, and the thin client interface [27]. These clouds are formed of a 7-layer architecture, consisting of layers: (1) as the layer of the physical infrastructure components, (2) layer of resources abstraction for virtualisation, (3) the layout layer for virtual Services, (4) the infrastructure as a service (IaaS) layer, (5) the layer for platform as a service (PaaS), (6) the application layer of software as a service (SaaS), and (7) the layer of applications for the tenants. The proposed multilayer security model may be compared to the cloud model of NIST [27] as follows. In NIST layers there are tenant users which could be hardware devices or virtual machines (VMs). Such a model may be implemented to each layer according to the principles of trustworthy computing [1,2].

Each session is aligned to layer six through a sequence of authentication and verification phases in the fourth and fifth layers. For applications which are hosted off-premises, layer seven access is made available via API interfaces. The presence of a firewall suggests infrastructure as a service (IaaS)[20], whereas management systems exist within the platform as a service (PaaS) layer. Software applications will reside in a software as a service (SaaS) tier.

3.1. Session Workflow

A typical session workflow is illustrated in Figure 1. The allocation of session IDs in layers three and two contributes to the setup of a new client by a future IIoT tenant user. This would be accompanied by the access identifier given in layer four. Following this stage, where the inspection of packets is a crucial task for each of the sessions that have taken place so far. The database of metadata (DB_{META}) and database of vault (DB_{VAULT}) layers require the verification of IIoT requests before the packet inspection is performed for each session using a database of intrusion prevention systems DB_{IPS} and database of anti-malware $DB_{ANTIMAL}$. The DB_{IPS} and DB_{META} link explicitly to PaaS functions within the context of the NIST model. In comparison, in the model Database of firewall DB_{FW} is known as IaaS. Supplementary authentication is required for each SaaS user, although at this stage, there is still a substantial number of verifications. Nevertheless, this verification is intended to enforce the company structure role-based permissions, such as the sub-set of employees, which offer access to the sensitive payroll information, for organisational data protection.

4. Hardware-Intrinsic Secure Multi-Layer Connectivity Model

The presented model considers users requesting access to services such as analytics in industrial infrastructure. Due to advancements in hardware technologies, users, as well as IIoT application services, incorporate hardware platforms on a large scale. One such advancement is the use of FPGAs solutions with hardware and software programmability providing flexibility and scalability to address IIoT requirements[28]. Applications such as data processing are an unavoidable part of IIoT, and

FPGAs are an invaluable part of meeting future processing demands. FPGA-based data centres provide a volume of computation and storage resources to be efficiently processed on the edge of the network.

The FPGA based accelerations have significant potential for industrial applications enabling real-time data processing by combining locally generated data with additional enterprise data. Hardware acceleration, flexibility, and performance provided by FPGAs are an attractive solution for 5G networks for meeting the changing and increasing demands of the wireless markets. Currently, FPGAs provide optimised solutions for 5G technologies such as cloud-based radio access network (cRAN), virtual radio access network (vRAN), Massive multiple-input, and multiple-output (MIMO), Backhaul, Fronthaul, Digital Radio Front-End [29].

With the rising number and connectivity of IIoT intelligent devices with the network, the model requires to process an increased volume of transactions. To deal with an increased processing volume, the multi-layer model requires compliance, where the proposed system dynamically provides the required flexibility and security. Below we describe the procedure adopted to introduce an IIoT device with an inbuilt design feature, which increases the level of security with the connecting components. We use the concept of hardware-intrinsic security, which develops security from the intrinsic properties of the silicon. The security primitive employed in this work is Physically-Unclonable Functions (PUF), which utilises intrinsic manufacturing differences in the electronic hardware for strengthening security.

We describe a protocol for secure connectivity in the network. The protocol introduces a series of steps that permit all new clients entering the IIoT system. To grant access to the IIoT system, a current client needs to introduce the new customer following a series of procedures (Algorithm-2), as described below. The model consists of K verification layers. Verification at each layer is assured using a PUF based security protocol. Every layer of the security model has a PUF. In the multilayered model $K = 7$, there exist a PUF for each existing user in each layer, which represents the fingerprint of every existing genuine member of the IIoT node. In this work, we use FPGAs for implementing the PUF. Packaged as a cloud manager, it generates a composite PUF and model (M_A), that represents the physical PUFs in the K layers of the model. Genuine clients receive an obfuscated bitstream consisting of a description of the mathematical PUF model through a secure communication channel. The genuine user introduced by an existing customer then downloads the bitstream and implements the PUF. We describe a PUF Based Authentication Protocol for verifying a client.

The cloud management plane handles the authorisation request initiated by the client U_A. The authorisation is processed by a security check involving the PUF, where q challenge sets(CH_p) of length n is sent to U_A together with a random number ($rand$). The received challenge bits are presented to the PUF model (M_A) at the client's end, and the corresponding responses are collected for every layer in the proposed model.As we are considering a K layer model, there reside K responses which represent a single challenge string for every layer. A pre-agreed shuffling scheme is used to scramble the entire responses ($K.q$) for all challenge sets. The client and the management plane accord with an encoding $E(.)$ and decoding $D(.)$ scheme sor secure transmission of PUF responses. The user U_A then sends the scuffled responses encoded with $E(.)$ to the cloud model for confirmation, and the clould management layer decodes with $D(.)$ to convert the response back and direct the responses to the respective cloud layer.

The original challenge bits of q are then added to the actual PUFs existing in the layers of the IoT cloud, and the responses are collected. The mathematical PUF and physical PUF responses are examined for a high similarity to declare the user U_A to be genuine.

The quality of the PUF is mainly determined by two parameters, which are reliability and security. A reliable PUF has a sufficiently long lifetime and provides a stable response under different external circumstances. Considering minuscule variations in XOR PUF responses, the presented algorithm provides a tolerance of 1% in PUF response comparison. The parameter security addresses the level of protection that a PUF offers against a wide range of attacks. We ensure security by employing a powerful Arbiter PUF with > 10 component XOR-PUFs stages to enhance security and to counter machine learning interventions [30].

4.1. Algorithm Design

To strengthen security, PUF based verification supplements the existing verification in the primary cloud multi-layer model. FPGAs residing in cloud layers contain PUFs describing all existing clients. Additionally, an existing genuine client comprises a mathematical model of PUF which is transferred from the cloud management unit. The mathematical model is implemented in the client FPGA following Dynamic Partial Reconfiguration. The mathematical model is constructed by the IIoT infrastructure using machine learning as it has access to internal parameters of constituent Arbitor PUF stages. To guaranty security, a strong PUF is employed in the system. A strong PUF promises to provide resistance to cloning by a malicious adversary adopting machine learning approaches [31]. This security is ensured by increasing the constituent Arbitor PUF stages to greater than 10.

Algorithm 1 describes the process to be followed to grant a request to access to an application. Various checks followed in the cloud model is represented in each step of the algorithm. The proposed model is flexible to incorporate additional security if required by provision to extend the security to additional layers. A database of previously used challenges is maintained by the cloud management unit to prevent repeated usage of same challenge bits and ensure security gained replay attacks.

Algorithm 1 Multi-layered security model using PUF: New client

Objective:

(a) The seven layer cloud model consisting of FPGA clouds verifies the identity of a new client FPGA (U_B) who is requesting access.
(b) The cloud model provides application access for the genuine client (U_B).

Prerequisites:

(a) New client $ClientU_B$, requesting application access is known to an existing client U_A as a genuine applicant
(b) $Cloud - FPGAs$ have built-in controllers to facilitate secure dynamic partial reconfiguration.
(c) $Client - FPGA$ has built-in controllers to facilitate secure dynamic partial reconfiguration initiated by the cloud.
(d) The $Cloud - FPGA$ fabric is divided into two parts, a) static fabric and b) dynamic fabric. Static fabric consists of hardware configurations which existed before deployment. The dynamic fabric of the $Cloud - FPGA$ is dedicated to configure additional security primitives (mostly PUFs) for any genuine clients using secure dynamic partial reconfiguration.
(e) The $client - FPGA$ fabric is divided into two parts, a) static fabric and b) dynamic fabric. Static fabric consists of hardware configurations which existed before deployment. The $Client - FPGA$ has secure remote DPR controllers in the static partition facilitating configuration of PUF mathematical model in the dynamic fabric, via an obfuscated bitstream.

Input:
P_{CT}, DB_{FW}, DB_{META}, DB_{VAULT}, DB_{IPS}, $DB_{ANTIMAL}$ of $UserU_A$

(a) Tenant session: S
(b) Contents of session packets:P_{CT}
(c) Contents of FW: DB_{FW}
(d) Contents of $TENANT_{META}$: DB_{META}
(e) Contents of $TENANT_{VAULT}$: DB_{VAULT}
(f) Contents of IPS : DB_{IPS}
(g) Contents of $ANTIMALWARE$: $DB_{ANTIMAL}$
Note: DB_j represents content DB of layer j

Output:
A value in Flag to show a successful dynamic partial reconfiguration ($Flag = 1$) or denied ($Flag = 0$).

Steps: Algorithm 1 continued.

1. Initialize $S = 1$, $E = 1$
2. U_i to management plane MP: request access to application A
3. MP to U_i: MP sends a random number $rand$ and a set of challenges CH_p consisting of q challenge bits each of length 'n'.
4. U_i calculates the following:
   - $Rim_{p,j} = Mi(CH_{p,j}), p = 1 \ldots q, j = 1 \ldots K$
   - $Rim = \{Rim_{p,j}, 1 \leq p \leq q, 1 \leq j \leq K\}$
   - CA_i=$S\Big(E(Rim), rand\Big)$
5. U_i to MP: certificate CA_i
6. **foreach** layer j **do**
   (a) Initialize $Mem = 0$, $Match = 0$
   (b) **If** $(E = 1)$
      (a) $MP : Rim_{p,j} = S'\Big(D(CA_i), rand\Big)$
      (b) MP to $Cloud - C_i$: Set of challenges CH_p and $Rim_{p,j}$
      (c) $Cloud - C_j$ calculates the following
         - $Rif_{p,j} = Pi(CH_{p,j}), p = 1 \ldots q, j = 1 \ldots K$
         - $N_{ij} = (1 - \frac{\sum_{(p=1)}^{q}(R_{imp} \oplus R_{ifp}}{q})$
         - **if** $Nij \geq 0.99$ $Mem = 1$
      (d) **if** $(P_{CT} \in DB_j, \mid DB_j \in \{DB_{FW}, DB_{META}, DB_{VAULT}\}$ AND $P_{CT} \notin DB_j, \mid DB_j \in \{DB_{IPS}, DB_{ANTIMAL}\})$; $Match = 1$
      (e) **if** $(Mem$ && $Match)$, $E = 1$; proceed to next higher layer
      (f) **else** Exit; set $E = 0$, $S = 0$; DenyTenantAccess()
7. **if** $S = 1$; AuthoriseTenantAccess()

The model contains physical PUF representing the client in each layer of the cloud model and mathematical PUF, describing the functionality of the physical PUF. An obfuscated bitstream is used to download the mathematical model of physical PUF using DPR. The mathematical model is constructed by the IIoT infrastructure using machine learning as it has access to internal parameters of constituent Arbitor PUF stages.

Robustness is provided by a strong PUF, which cannot be cloned by malicious third parties. The requirements for the PUF considered in this work include (A) a Strong PUFs with a vast number of possible challenges, (B) unpredictability of challenge responses which means the difficulty to extrapolate or predict the CRPS from the known CRPs.

The security is ensured by increasing the constituent Arbitor PUF stages to greater than 10. Algorithm 2 provides authentication for all user requests for entry to the IIoT system. Each step of the process delivers security checks required at each stage of the layer. The algorithm is briefly described below. A set of challenges are generated, which excludes prior sets, and these are used between client and cloud layers during their authentication interactions. Each authentication requires a collection of q challenge bits, each being length n. Both the mathematical model and the physical model are provided with the same to generate the responses. At each cloud layer, the produced responses of the mathematical PUF and the physical PUF are compared for verification. The high similarity of responses (>= 99%) is considered genuine, and the client is granted to proceed to the next layer of security checks. A database of previously used challenges bits is maintained to disregard any repeated usage, which would otherwise provide a chance for a replay attack. For a challenge set size of q and

length n-bit used for each authentication attempt, provides $(2^n/q)$ possible attempts of access on the application. The challenge bit-size is extensively large, requiring billions of years to be completely exhausted.

Algorithm 2 Multi-layered security model using PUF: Client is an existing User [2]

Objective:

(a) The seven layer cloud model consisting of FPGA clouds verifies the identity of a client FPGA (U_A) who is requesting access.
(b) The cloud model provides application access for the genuine client (U_i).

Prerequisites:

(a) An n-bit input, 1-bit output XOR PUF P_1 is reconfigured in all layers of the $Cloud - FPGA$. There exists a PUF for every authenticated user. PUF P_{ij} represents the identity of the user i in the cloud layer j.
(b) A combined mathematical model M_i representing all the K PUFs in the cloud layers, resides with each user U_i.
(c) $Cloud - FPGA$ and user U_i have agreed on a fixed encoding scheme $E()$ and a decoding scheme $D(.)$, such that for any binary string x, $E(.)$ and $D(.)$ are injective, $X = E(x)$ and $D(X) = x$.
(d) $Cloud - FPGA$ and user U_i have agreed on a shuffling scheme $Y = S(X, rand)$, and $S'(Y, rand) = X$ where $rand$ is a random number.

Input:

S, P_{CT}, DB_{FW}, DB_{META}, DB_{VAULT}, DB_{IPS}, $DB_{ANTIMAL}$

(a) Tenant session: S
(b) Contents of session packets:P_{CT}
(c) Contents of FW: DB_{FW}
(d) Contents of $TENANT_{META}$: DB_{META}
(e) Contents of $TENANT_{VAULT}$: DB_{VAULT}
(f) Contents of IPS : DB_{IPS}
(g) Contents of $ANTIMALWARE$: $DB_{ANTIMAL}$
Note: DB_j represents content DB of layer j

Output:

A value in variable S to show that the application access is granted ($S = 1$) or denied ($S = 0$).

Additionally, our model is fully flexible with DPR capability, which permits new security primitives, including novel PUF architectures, to be replaced with the existing once. This further increases the life span of the model. In addition, considering the research in the area [7], the possibility of repeated challenges occurring is highly unlikely for comparable challenge set volumes.

The presence of new users being proposed by an pre-existing authenticated system users is made possible by the sharing of model responses. Once the existing client has been successfully authenticated, new PUFs are created by the FPGA at run-time. After DPR, the mathematical PUF model is downloaded to the new user FPGA using an obfuscated bitstream. Again, the security model makes the assumption that the security process of maintaining system integrity is followed by a secure DPR process. The new cloud user will then use an algorithm of 1 to obtain the application layer.

4.2. Device Parameter Analysis of Client FPGAs

A genuine client, which turns to be potentially malicious by modifying the client device architecture to attack the IIoT application, is confirmed by client device parameter verification using neural networks. The client analysis process strengthens the security of the IIoT application against potentially malicious clients. A legitimate cloud service requires each IIoT client FPGA to satisfy specific requirements. Firstly, the FPGAs require a Dynamic Partial Reconfiguration (DPR) ability, which facilitates the set up of PUF primitives in the FPGA fabric. DPR allows dynamic reconfiguration of hardware units in selected regions on the FPGA framework.

Steps: Algorithm 2 continued.

1. Initialize $V = 1$, $E = 1$, $Flag = 0$
2. U_B requests U_A, for an introduction to access application A
3. U_A to MP: request introduction of U_B to cloud layers C_j
4. MP to U_A: MP sends a random number $rand$ and a set of challenges CH_p consisting of q challenge bits each of length 'n'.
5. U_A calculates the following:
   - $RAm_{p,j} = MA(CH_{p,j})$, $p = 1 \ldots q$, $j = 1 \ldots K$
   - $RAm = \{RAm_{p,j}, 1 \leq p \leq q, 1 \leq j \leq K\}$
   - CA_A=$S\Big(E(RAm), rand\Big)$
6. U_A to MP: certificate CA_A
7. **foreach** layer j **do**
   (a) Initialize $Mem = 0$, $Match = 0$
   (b) **If** $(E = 1)$
      (a) $MP : RAm_{p,j} = S'\Big(D(CA_A), rand\Big)$
      (b) MP to $Cloud - C_j$: Set of challenges CH_p and $RAm_{p,j}$
      (c) $Cloud - C_i$ calculates the following
         - $RAf_{p,j} = PA(CH_{p,j})$, $p = 1 \ldots q$, $j = 1 \ldots K$
         - $N_{Aj} = (1 - \frac{\sum_{(p=1)}^{q} (R_{Amp} \oplus R_{Afp}}{q})$
         - **if** $NAj \geq 0.99$ $Mem = 1$
      (d) **if** $(P_{CT} \in DB_j, \mid DB_j \in \{DB_{FW}, DB_{META}, DB_{VAULT}\}$ AND $P_{CT} \notin DB_j, \mid DB_j \in \{DB_{IPS}, DB_{ANTIMAL}\})$; $Match = 1$
      (e) **if** $(Mem$ && $Match)$, $E = 1$; proceed to next higher layer
      (f) **else** Exit; set $E = 0$, $Flag = 0$
8. **if** $V = 1$; Verified introducing client
   (a) **foreach** layer j **do**
      (a) $Cloud - FPGA$, C_j initiates DPR and configures a new PUF $P_{B,j}$, PUF $P_{B,j}$ represents the identity of the U_B in the cloud layer j
      (b) C_j to MP PUF modeling parameters $param_j$
   (b) MP generates a combined Mathematical model M_B of all PUFs $P_{B,j}$ in the cloud layers
   (c) MP generates obfuscated bitstreams of PUF mathematical model M_B
   (d) MP initiates remote dynamic partial reconfiguration of PUF M_B in the dynamic partition of the $client - FPGA$ U_B
   (e) $Flag = 1$ and exit; follow protocol-1. U_B is same as any other existing client.

The FPGA floor plan requires dynamic partitions that promote analysis of the fabric by the cloud service. Although DPR offers tremendous flexibility for IIoT applications, security needs to be ensured to avoid DPR based Trojan insertions as proven in [7]. An additional security measure adopted in

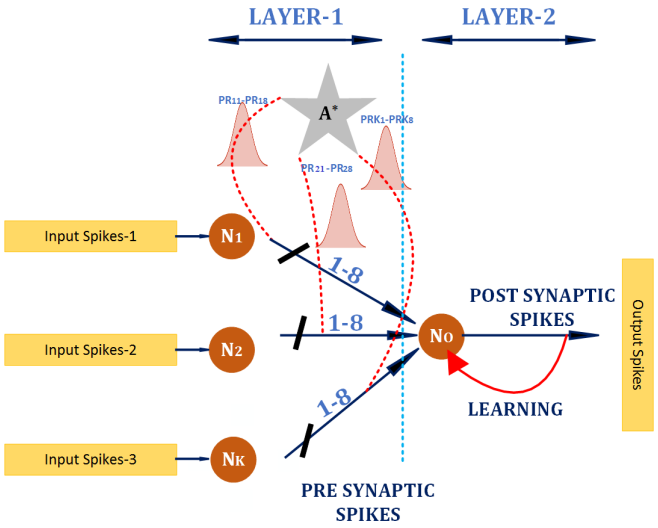

Figure 2. Parameter verification using spiking neural network

the proposed scheme is to analyse the device parameters of the client FPGA. DPR performs this by sending an obfuscated and downloadable bitstream which collects the client device parameters. The device parameters are tested at the malware detection layer to assure that variations in the attributes of the FPGA clients. A client FPGA signature is a mechanism for identifying malicious adversaries. An initial DPR process implements a design that collects the device parameters, and a second DPR process erases the downloaded bitstreams. These device parameters are directly collected by the cloud management unit to evade manipulations from the client.

The proposed architecture for device parameter verification for Trojan analysis is shown in Figure 2. The design consists of two layers of neurons, where the input layer neurons (LAYER-1), produces a spike train with a frequency proportional to the device parameter. The number of input layer neurons represents to the number of device attributes that are analysed. The second layer neuron responds based on the spike rate received from its presynaptic neurons. In Figure 2, K input layer neurons are shown. There exist eight parallel connections between two tiers of pre-post synaptic neurons. This is to mimic the parallel connection between neurons in the brain-inspired systems, which aids in building post-synaptic potential and enhances fault-tolerance. The pattern identification procedure regulates the spike rate between layer 1 and 2. This is depicted using the Gaussian distribution shown between the neurons.

The distribution represents a variable transmission probability depending on the particular pattern. The nomenclature PR_{Kr} represents transmission probability between presynaptic neuron K and the post-synaptic neuron in the r^{th} interconnection between the pair. The output layer neuron provides a stable enable signal for the client FPGA if the received device parameters are within scope. This principle of using a spiking neural network is derived from [32,33], and hardware realization of the approach is described in [34].

However, in [32–34], the authors derive bio-inspired principles for homeostasis targeting robotic applications, where this paper emphasis the use of similar methodologies for hardware Trojan detection. Bio-inspired computing develops computational models using various models of biology. Brain-inspired computing is a subset of bio-inspired computing, which is mainly based on the mechanism of the brain. Brain-inspired models help to narrow the hardware Trojan detection process based on the mechanism of the brain, which produces a compact computational model rather than the complex biological process involved in the former. A pattern identification protocol verifies the

pattern where spike to the postsynaptic neuron (LAYER-2) is regulated using a transmission regulation following a Gaussian relation. A high transmission probability (PR) is provided by the transmission regulation unit provided the device parameters are in the acceptable range.

A lower PR indicates a more significant deviation from the device parameter standards and fewer input spikes arriving at LAYER-2, which provides a stable firing rate by following Spike-timing-dependent plasticity (STDP) [35] and Bienenstock–Cooper–Munro (BCM) learning rules [36] for spike rates in the permissible range. Otherwise, the postsynaptic spikes drop to zero. Multiple connections are laid between each pre-post synaptic neurons to increase the security of the detection unit from any intruder from attacking the Trojan analysis unit.

5. Experimental Results

In this work, we implemented an XOR PUF construct consisting of 10 parallel Arbiter PUFs with 64 switch blocks on Xilinx Nexys 4 DDR board with Artix-7 FPGA (device xc7a100t, package csg324, speed -1) [37]. Verilog Hardware Description Language (HDL) is used for design purposes, and Electronic design automation (EDA) Xilinx ISE 14.7 design suite [38]. For further analysis, we used the Xilinx power analysis tool and Chipscop-Pro [39].

The implementation cost of the PUF design is shown in Table 1. The design used only a fraction (8%) of the FPGA slice of the device (Artix-7 FPGA), which is negligible for large FPGAs stationed for high-end applications. Table 1, reports the size of bitstream required to reconfigurable PUF, which is relatively small. A difference based partial reconfiguration methodology is used for PUF reconfiguration over the network [40]. Additionally, new FPGA tools (Partial Reconfiguration flow in Vivado Design Suite) provided specific flow implementations for dynamic partial reconfiguration. In this work, an 8 bit configuration was followed for the Internal Configuration Access Port (ICAP) and a clock rate of $100MHz$. The above settings enabled DPR to be completed in microseconds, which proved to be a real-fit for cloud-based applications.

The proposed device parameter verification circuitry is shown in figure 2 using a Xilinx Nexys 4 DDR board, Artix-7 FPGA (device xc7a100t, package csg324, speed -1). Neural activities were monitored using an *Integrated Logic Analyzer* (ILA). The *Xilinx Power Estimation and Analysis Tools* and *Timing Closure and Design Analysis* are used additionally. We report hardware and power footprints in Table 2.

Table 1. Implementation Overhead [2]

Hardware Consumption *	**Slice 1291**	**Slice Reg 10**	**LUTs 1282**
Power Consumption			0.082W
Bitstream Size			3737KB

*Note The design does not contain any LUTRAMs, BRAMs/FIFOs, DSPs or buffers

Table 2. Implementation Overhead of the device parameter verification unit presented in Figure 2

Parameter/Components Hardware Consumption	**Slice 14471**	**Slice Reg 33707**	**LUT 25065**	**DSP 30**
Total On-chip Power	0.082W			

The Gaussian function representing transmission probability between LAYER-1 and LAYER-2 was implemented using linear approximations to minimize the hardware overhead. The neurons used were deployed using Leaky-Integrate and Fire(LIF) neuron models [41], as they are computationally efficient for hardware implementations. Hardware utilisation and synapses increased, which operated based on a BCM-STDP rule. Alternate synaptic rules such as Spike Driven Synaptic Plasticity (SDSP) [42]-based

synaptic rule will be the focus of future investigations. These have the potential to lower synaptic weight from $32-bits$ to perform $1-bit$ operations. Assessing the practicality of the proposed system in real-world scenarios[43], the overall performance of the system was explored through a Python/SimPy [44] simulation model.

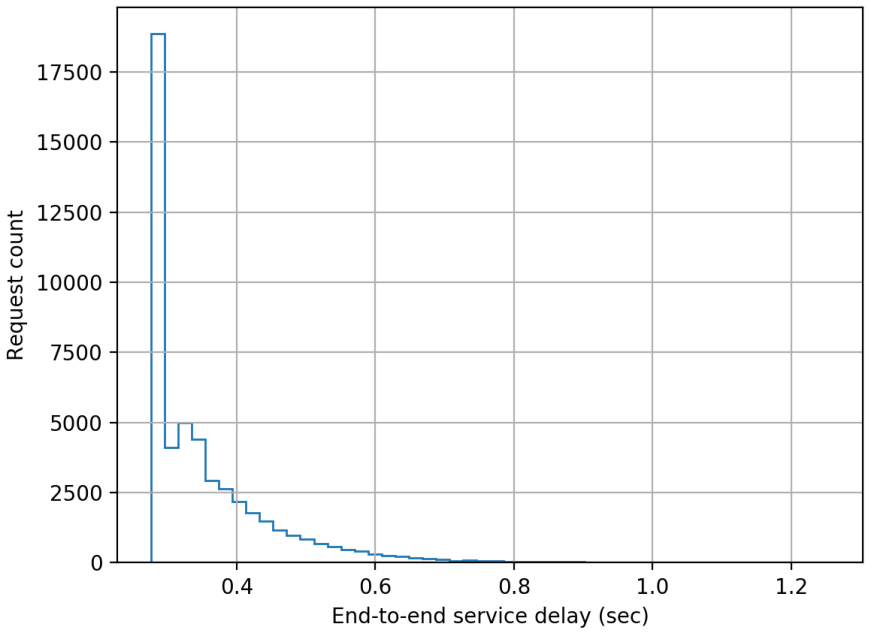

Figure 3. Distribution of end-to-end service delay at a mean request rate of 10 requests-per-second

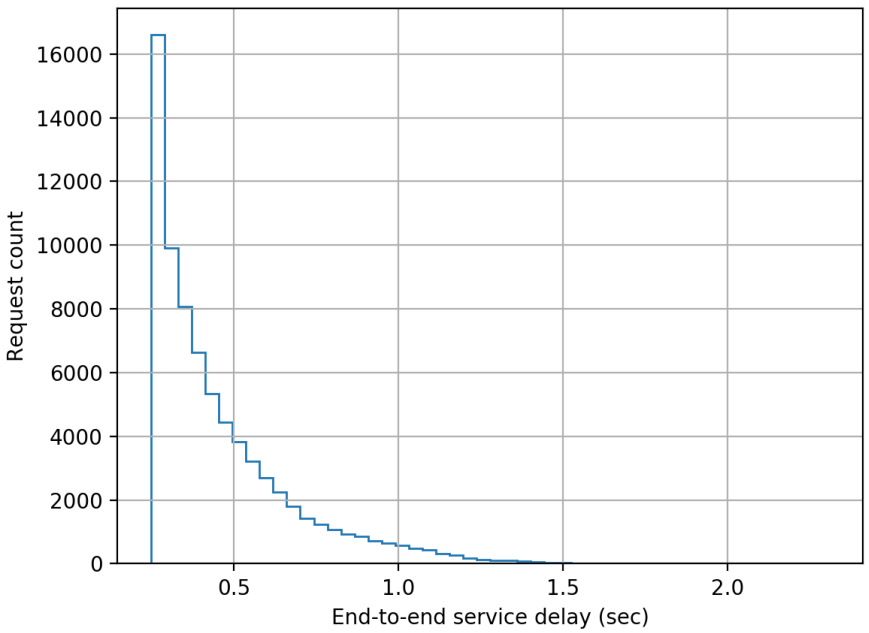

Figure 4. Distribution of end-to-end service delay at a mean request rate of 15 requests-per-second

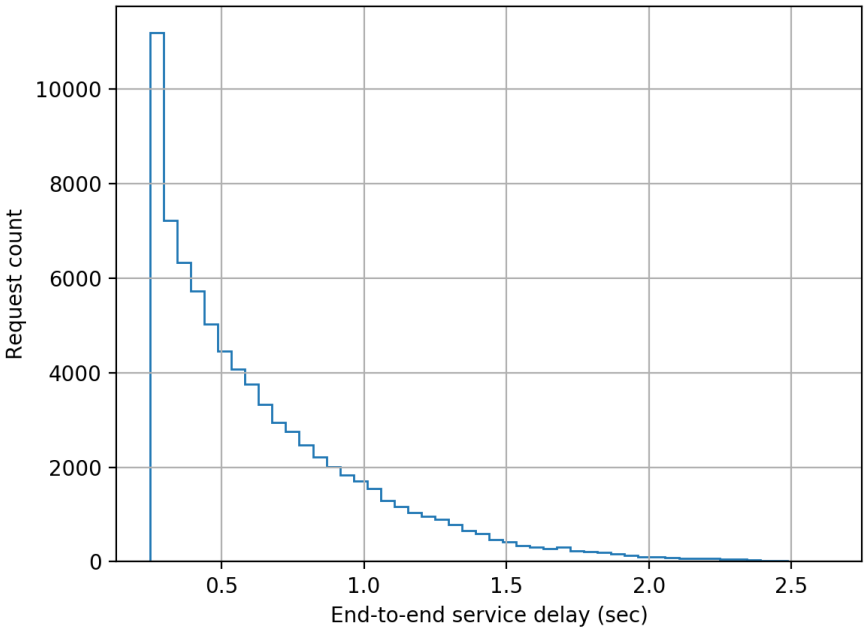

Figure 5. Distribution of end-to-end service delay at a mean request rate of 16 requests-per-second

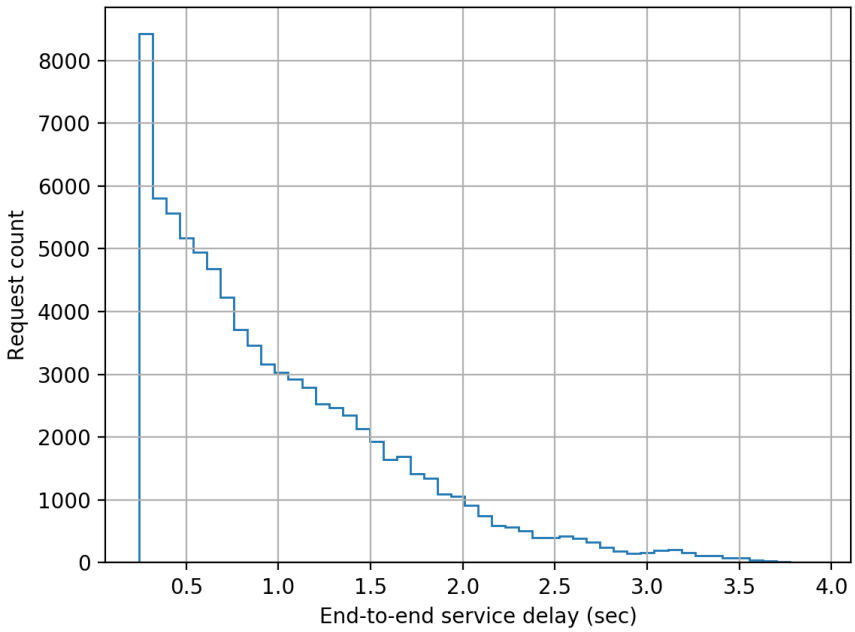

Figure 6. Distribution of end-to-end service delay at a mean request rate of 17 requests-per-second

The model developed comprised of five cascaded servers as per the scheme described earlier in this paper, and each server was configured to have a log-normally distributed processing delay with a mean delay of 50 ms (standard deviation = 10 ms). The simulation was run for 1000 s of simulation time for each separate load on the system measured as the number of authentication requests submitted per

second (RPS). The authentication requests were modelled as having a Poisson distribution. End-to-end delay data was collected for each authentication transaction in a run, and the following figures present histograms of the delays. Histograms were chosen as for a user of the system the mean or median delay only presents a very limited view of the actual performance that will be delivered to an individual user. The adoption of histograms to show the distribution of delays provides users with a far better indication of the range of performance that they will experience across a large number of authentication requests.

From Figure 3 to Figure 8 as presented here, some conclusions about the performance of the system can be drawn. The definition of acceptable performance in terms of authentication delay is, of course, subjective. For the sake of this discussion we will take a somewhat arbitrary position that if the overwhelming majority of requests are serviced in less than 1 s then performance is deemed acceptable.

At low loading (10 RPS), all requests are serviced within our 1 sec limit, while at a higher rates of 15 RPS and 16 RPS, progressively more requests take longer than our 1 s target, but overall performance could still be deemed acceptable.

However, at 17 RPS a very significant fraction of requests take longer than 1 s to service, with the some having to wait over 3 s. At 19 RPS, the system effectively fails, and cannot cope with the volume of traffic. This is consistent with expectations as a cascade of five stages with a fixed service time of 50 ms each per request should cope with 20 uniformly distributed RPS with an end-to-end delay of 1.25 s. The breakdown in performance in the simulated system is due to the randomness in timing of request generation and the randomness in processing time at each node.

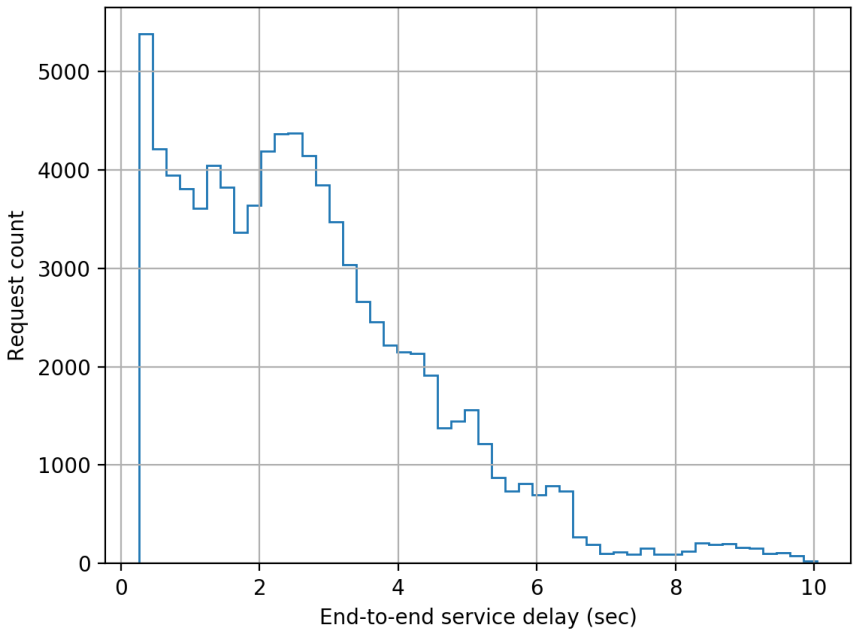

Figure 7. Distribution of end-to-end service delay at a mean request rate of 18 requests-per-second

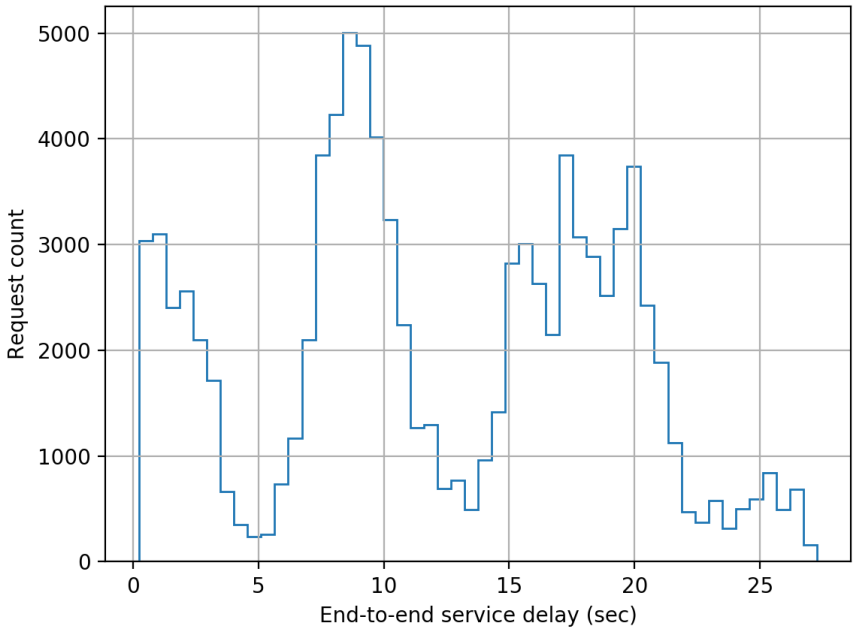

Figure 8. Distribution of end-to-end service delay at a mean request rate of 19 requests-per-second

6. Conclusions

This article we extends previous work on implementing multiple layers in cloud security. We add hardware security primitives and trojan detection units in combination, for a robust, multi-layer security architecture. The proposed work describes a PUF-based system with a brain-inspired device parameter analysis unit that demonstrates the ability for attack prevention from both external and internal attacks of interest primarily in the IIoT context. A vast array of surreptitious activity may be isolated to withstand a number of attack vectors, by considering security at every cloud- abstraction layers. Multiple layers of packet inspection secure the IIoT application against other opponents who may concurrently implement many tools and approaches to compromise the system security.

The security is primarily maintained by PUF-based security protocols that rely on unique device fingerprints which are hard to be compromised. The inherent flexibility and scalability provided by DPR capability with plug-and-play of new security primitives is a vital advantage of the proposed approach providing a promising direction for 5G enabled IIoT. The DPR facility removes constraints of security functions, which later is replaceable with the more secure ones.

Additionally, the hardware device parameter inspection avoids further attacks on the IIoT application using parameter variations. The continued expansion and accessibility of IIoT hardware requires flexible hardware programmability as provided by novel FPGAs. In addition, embedding analytics functions into industrial organisations requires high computational capabilities and flexible architectures provided by FPGAs. To provide satisfactory operations for the communication demand, IIoT devices with high-speed 5G networking technology is a requirement.

At the same time, appreciating the flexibility, security cannot be compromised in the infrastructure. In order to guarantee security within the model, we proposed to use primitive hardware security, for example, PUFs. The monitoring of client IIoT side-channel parameters also enhances security. In the IIoT cloud era, all software and hardware innovations have to operate together to ensure better security; failing to function might be catastrophic.

Author Contributions: The work described in this article is a collaborative effort from all of the authors. Conceptualisation, A.P.J., H.A., and R.H.; Methodology, H.A., P.L., and RH; design, implemention, and generation

of results on hardware—FPGA, A.P.J., P.L., and H.A.; analysis and interpretation of results, H.A., R.H., P.L., and T.A.; preparing the draft, review, and editing R.H., T.A., H.A. All authors reviewed the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Alrawais, A.; Alhothaily, A.; Hu, C.; Cheng, X. Fog computing for the internet of things: Security and privacy issues. *IEEE Int. Comput.* **2017**, *21*, 34–42.
2. Al-Aqrabi, H.; Johnson, A.P.; Hill, R.; Lane, P.; Liu, L. A Multi-layer Security Model for 5G-Enabled Industrial Internet of Things. In Proceedings of the International Conference on Smart City and Informatization, Gungzhou, China, 12–15 November **2019**; pp. 279–292.
3. Alsboui, T.; Qin, Y.; Hill, R. Enabling Distributed Intelligence in the Internet of Things Using the IOTA Tangle Architecture. In Proceedings of the 4th International Conference on Internet of Things, Big Data and Security, Heraklion, Greece, 2–4 May **2019**; pp. 392–398
4. Beer, M.D.; Hill, R.; Huang, W.; Sixsmith, A. An agent-based architecture for managing the provision of community care: The INCA (Intelligent Community Alarm) experience. *AI Communicat.* **2003**, *3*, 179–192.
5. Al-Aqrabi, H.; Hill, R. Dynamic multiparty authentication of data analytics services within cloud environments. In Proceedings of the 2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems, Exeter, UK, 28–30 June **2018**; pp. 742–749.
6. Sotiriadis, S.; Bessis, N.; Antonopoulos, N.; Hill, R. Meta-scheduling Algorithms for Managing Inter-cloud Interoperability. *Int. J. High Perform. Comput. Netw.* **2013**, *7*, 156–172.
7. Johnson, A.P.; Chakraborty, R.S.; Mukhopadhyay, D. A PUF-enabled secure architecture for FPGA-based IoT applications. *IEEE Transact. Multi Scale Comput. Syst.* **2015**, *1*, 110–122.
8. Narasimhan, S.; Du, D.; Chakraborty, R.S.; Paul, S.; Wolff, F.G.; Papachristou, C.A.; Roy, K.; Bhunia, S. Hardware Trojan detection by multiple-parameter side-channel analysis. *IEEE Transact. comput.* **2012**, *62*, 2183–2195.
9. Díaz-Sánchez, D.; Almenarez, F.; Marín, A.; Sánchez-Guerrero, R.; Arias, P. Media Gateway: bringing privacy to private multimedia cloud connections. *Telecommunicat. syst.* **2014**, *55*, 315–330.
10. Pearson, S. Taking account of privacy when designing cloud computing services. In Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, Vancouver, Canada, 23–23 May, **2009**; pp. 44–52.
11. Luo, S.; Lin, Z.; Chen, X.; Yang, Z.; Chen, J. Virtualization security for cloud computing service. In Proceedings of the 2011 International Conference on Cloud and Service Computing, Hong Kong, China, 12–14 December **2011**; pp. 174–179.
12. Semenko, Y.; Saucez, D. Distributed privacy preserving platform for ridesharing services. In Proceedings of the International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage, Atlanta, GA, USA, 14–17 July 2019; pp. 1–14.
13. Alsboui, T,; Qin, L.; Hill, R.; Al-Aqrabi, H. Enabling Distributed Intelligence in the Internet of Things with IOTA and Mobile Agents. *Comput. Springer Wien.* **2020**, *1*, 28.
14. Roman, R.; Najera, P.; Lopez, J. Securing the internet of things. *Computer* **2011**, *44*, 51–58.
15. Li, J.; Zhang, Y.; Chen, Y.F.; Nagaraja, K.; Li, S.; Raychaudhuri, D. A mobile phone based WSN infrastructure for IoT over future internet architecture. In Proceedings of the 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, Beijing, China , 20–23 August 2013; pp. 426–433.
16. Hill, R.; Polovina, S.; Beer, M.D. From Concepts to Agents: Towards a Framework for Multi-Agent System Modelling. In Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 05), Utrecht, The Netherlands, 25–29 July 2005; pp. 1155–1156.
17. Polovina, S.; Hill, R.; Beer, M.D. Enhancing the Initial Requirements Capture of Multi-Agent Systems through Conceptual Graphs. In Proceedings of the Thirteenth International Conference on Conceptual Structures (ICCS2005): Conceptual Structures at Work, Springer Lecture Notes in Artificial Intelligence (LNAI), Marburg, Germany, 1–4 July 2005; pp. 439–452.

18. Hill, R.; Polovina, S.; Shadija, D. Transaction Agent Modelling: From Experts to Concepts to Multi-Agent Systems. In Proceedings of the Fourteenth International Conference on Conceptual Structures (ICCS2006): Conceptual Structures: Inspiration and Application, Springer Lecture Notes in Artificial Intelligence (LNAI), Paris, France, 20–22 August 2014; pp. 247–259.
19. Demchenko, Y.; Ngo, C.; de Laat, C.; Lopez, D.R.; Morales, A.; García-Espín, J.A. Security infrastructure for dynamically provisioned cloud infrastructure services. In *Privacy and Security for Cloud Computing*; Springer: Berlin, Germany, 2013; pp. 167–210.
20. Carvalho, M. Secaas-security as a service. *ISSA J.* **2011**, pp. 20–24. Available online: https://www.researchgate.net/publication/286650384_Towards_Security_as_a_Service_SecaaS_On_the_modeling_of_Security_Services_for_Cloud_Computing (accessed on 29 March 2020).
21. Mell, P.; Grance, T. The NIST definition of cloud computing. Available online: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf (accessed on 29 March 2020).
22. Shadija, D.; Rezai, M.; Hill, R. Towards an Understanding of Microservices In Proccedings of the 23rd International Conference on Automation and Computing (ICAC2017), Huddersfield, UK, 7–8 September 2017; pp. 1–6
23. Sanchez-Iborra, R.; Sánchez-Gómez, J.; Pérez, S.; Fernández, P.J.; Santa, J.; Hernández-Ramos, J.L.; Skarmeta, A.F. Enhancing lorawan security through a lightweight and authenticated key management approach. *Sensors* **2018**, *18*, 1833.
24. Taheri, S.; Yuan, J.S. A cross-layer biometric recognition system for mobile IoT devices. *Electronics* **2018**, *7*, 26.
25. Al Aqrabi, H.; Liu, L.; Hill, R.; Antonopoulos, N. A multi-layer hierarchical inter-cloud connectivity model for sequential packet inspection of tenant sessions accessing BI as a service. In Proceedings of the 2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst, Paris, France, 20–22 August **2014**; pp. 498–505.
26. Al-Aqrabi, H,; Hill, R,; Lane,; P,; Aagela, H. Securing Manufacturing Intelligence for the Industrial Internet of Things, In Proccedings of the Fourth International Congress on Information and Communication Technology, London, UK, 27–28 February 2019; pp. 267—282.
27. Badger, L.; Bohn, R. NIST US Government cloud computing technology roadmap. *Release* **2011**, *1*, 500–293.
28. Baker, C.; Anjum, A.; Hill, R.; Bessis, N.; Kiani, S.L. Improving Cloud Datacentre Scalability, Agility and Performance Using OpenFlow. In Proceedings of the Fourth International Conference on Intelligent Networking and Collaborative Systems (InCoS2012), Bucharest, Romania, 19–21 September 2012; pp. 20–27.
29. Nikaein, N.; Schiller, E.; Favraud, R.; Knopp, R.; Alyafawi, I.; Braun, T. Towards a cloud-native radio access network. In *Advances in mobile cloud computing and big data in the 5g era*; Springer: Berlin, Germany, **2017**; pp. 171–202.
30. Pappu, R.; Recht, B.; Taylor, J.; Gershenfeld, N. Physical one-way functions. *Science* **2002**, *297*, 2026–2030.
31. Anju, P,; Al-Aqrabi, H,; Hill, R. Bio-Inspired Approaches to Safety and Security in IoT-Enabled Cyber-Physical Systems, *Sens. Multidisciplinary Digital Publish. Instit.* **2020**, *20*, 844.
32. Liu, J.; McDaid, L.J.; Harkin, J.; Karim, S.; Johnson, A.P.; Millard, A.G.; Hilder, J.; Halliday, D.M.; Tyrrell, A.M.; Timmis, J. Exploring self-repair in a coupled spiking astrocyte neural network. *IEEE transact. neural networks learn. syst.* **2018**, *30*, 865–875.
33. Liu, J.; McDaid, L.J.; Harkin, J.; Wade, J.J.; Karim, S.; Johnson, A.P.; Millard, A.G.; Halliday, D.M.; Tyrrell, A.M.; Timmis, J. Self-repairing learning rule for spiking astrocyte-neuron networks. In Proceedings of the International Conference on Neural Information Processing, Guangzhou, China, 14–18 November 2017; pp. 384–392.
34. Johnson, A.P.; Liu, J.; Millard, A.G.; Karim, S.; Tyrrell, A.M.; Harkin, J.; Timmis, J.; McDaid, L.; Halliday, D.M. Fault-tolerant Learning in Spiking Astrocyte-Neural Networks on FPGAs. In Proceedings of the 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), Pune, India, 6–10 January 2018; pp. 49–54.
35. Dan, Y.; Poo, M.m. Spike timing-dependent plasticity of neural circuits. *Neuron* **2004**, *44*, 23–30.
36. Bienenstock, E.L.; Cooper, L.N.; Munro, P.W. Theory for the development of neuron selectivity: orientation specificity and binocular interaction in visual cortex. *J. Neurosci.* **1982**, *2*, 32–48.

37. Digilent Inc, Nexys A7 FPGA Board Reference Manual. Available online: reference.digilentinc.com/_media/reference/programmable-logic/\nexys-a7/nexys-a7_rm.pdf. (accessed on 19 March 2020).
38. Xilinx Inc, XilinxISE: ISE Design Suite Overview. Available online: www.xilinx.com/support/documentation/sw_manuals/xilinx11/ise_c_overview.htm. (accessed on 19 March 2020).
39. Xilinx Inc, ChipScope Pro and the Serial I/O Toolkit. Available online: www.xilinx.com/products/design-tools/chipscopepro.html. (accessed on 19 March 2020).
40. Gören, S.; Turk, Y.; Ozkurt, O.; Yildiz, A.; Ugurdag, H.F. Achieving modular dynamic partial reconfiguration with a difference-based flow. In Proceedings of the ACM/SIGDA international symposium on Field programmable gate arrays, New York, NY, USA, 15 July 2013; p. 270.
41. Gerstner, W.; Kistler, W.M. Spiking neuron models: Single neurons, populations, plasticity; Available online: https://icwww.epfl.ch/~gerstner/SPNM/SPNM.html (accessed on 29 March 2020).
42. Fusi, S.; Annunziato, M.; Badoni, D.; Salamon, A.; Amit, D.J. Spike-driven synaptic plasticity: theory, simulation, VLSI implementation. *Neural computat.* **2000**, *12*, 2227–2258.
43. García-Campos, J.M.; Reina, D.G.; Toral, S.L.; Bessis, N.; Barrero, F.; Asimakopoulou, E; Hill, R. Performance Evaluation of Reactive Routing Protocols for VANETs in Urban Scenarios Following Good Simulation Practices. In Proceedings of the 9th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Blumenau, Brazil, 8–10 July 2015; pp.1–8.
44. Müller, K; Vignaux, T.; Lünsdorf, O.; Scherfke, S. SimPy v2.2 documentation. Available online: https://pythonhosted.org/SimPy/Manuals/HISTORY.html (accessed on 29 March 2020).