International Journal of Information Security manuscript No.
(will be inserted by the editor)

Dynamic Authentication for Intelligent Sensor Clouds in the Internet of Things

Hussain Al-Aqrabi · Ahmed M. Manasrah · Richard Hill · Mohammed Q. Shatnawi · Mohammad Sh Daoud · Hoda Alkhzaimi

Received: date / Accepted: date

Abstract Sensor clouds are formed by IP-enabled wireless sensors and Internet of Things (IoT) devices that are used for sensing and actuation in commercial and industrial applications. Data collected by the sensors are consolidated by distributed cloud data consolidation (DCS) servers to be utilized as raw sensory information by applications running data analytics and actuation functions. Alternatively, DC servers may feed sensor data to the cloud-hosted Big Data Analytics (BDS) servers. Sensor clouds and their respective DCS servers, as well as BDS servers, may form different security realms. These security realms' ownership structures are complicated and differ from standard database servers, necessitating a dependable authentication technique to provide trusted access to DC and BDS servers. This paper proposes a new Multiparty authentication framework to authenticate applications requesting access to the DCS and BDS servers without direct human or application access to the sensors and actuators. Only DC servers are permitted to communicate with sensors/actuators, and only applications

Hussain Al-Aqrabi · Ahmed M. Manasrah
Department of Computer Information Science, Higher Colleges of Technology, Shariah, UAE
E-mail: halaqrabi,amanasrah@hct.ac.ae

Richard Hill
Department of Computer Science, University of Huddersfield, Huddersfield, UK
E-mail: r.hill@hud.ac.ae

Mohammed Q. Shatnawi
Faculty of Computer and Information Technology, Jordan University of Science and Technology, Irbid, Jordan,
E-mail: mshatnawi@just.edu.jo

Ahmed M. Manasrah
Computer Sciences Department, Yarmouk University, Irbid, Jordan
E-mail: ahmad.a@yu.edu.jo

Mohammad Sh Daoud
College of Engineering, Al Ain University, Abu Dhabi, UAE
E-mail: mohammad.daoud@aau.ac.ae

Hoda Alkhzaimi
Engineering Department and Research Institute, New York University Abu Dhabi, UAE
E-mail: hoda.alkhzaimi@nyu.edu

certified by a Session Authority Cloud (SAC) are granted access to DCS/BDS servers via an authentication protocol that includes many information and key exchanges. This solution may assure the reliable deployment of sensor clouds in different critical application domains (i.e. industry, commercial, national security, and defense, etc while reducing the potential of direct espionage of sensed/actuated systems. Linear Temporal Logic (LTL) is used to explicitly analyze and establish the correctness of the presented framework. OPNET modeling and simulations are used to illustrate the protocol's design and operations. The results demonstrate that multi-party authentication is conceivable for Sensor cloud computing systems.

Keywords Inter-cloud · Multi-cloud · Sensor cloud · Authentication · Cloud Security · Internet of Things · Big Data Analytics

List of Abbreviations

ID_r^A	Cloud membership root key of user A.
ID_s^A	Sub-domain membership key of user A.
SAC_{DB}	DB Database.
SAC_{SH}	Session handler.
BDS	Big Data Analytics.
DCS	Distributed cloud data consolidation.
LTL	Linear Temporal Logic.
MPSH	Multi-party session handler.
RFID	Radio frequency identification.
SAC	Session Authority Cloud.
SSO	Single sign-on.
WSN	Wireless sensor networking.

1 INTRODUCTION

The Internet of Things (IoT) is a global network of diverse smart objects, such as environmental and wearable sensors, uniquely identified and accessible through network technologies like WiFi and Bluetooth. This interconnected system offers promising applications for enhancing various aspects of life. In 2020, predictions estimated 50 billion connected "things," including sensors, smartphones, and GPS devices [1,2]. While this projection was ambitious, the current reality is not far off. Forecasts indicate that by 2030, the IoT ecosystem will significantly expand to approximately 125 billion connected devices [3]. This growth introduces challenges related to data volume, speed, scalability, and, notably, authentication, which is the primary focus of this work.

As IoT networks rapidly become one of the fastest-growing industries, there is a notable focus on Industrial IoT (IIoT). The challenge in IIoT lies in adapting to the diverse existing industrial infrastructures and devices [1]. Moreover, the introduction of new business models that incorporate IoT and IIoT devices raises legitimate security and privacy concerns. Many of these devices operate seamlessly without geographic restrictions, connecting across users, industry, and government boundaries. IoT devices possess the capability to generate, analyze, and respond to their environment. The applications of IoT provide services that contribute to improving people's lives. Nevertheless, this convenience comes at a potential cost in terms of privacy and security [4]. The inability of IoT Manufacturers failure to implement robust security measures on devices has raised concerns among security specialists about the potential risks associated with the Internet connecting numerous unsecured devices [3]. The evolving demands of IoT, where allocating resources for secure data exchange may reach theoretically unattainable levels, prompting a need for an effective solution. FOG/EDGE computing, a distributed architecture, directly addresses this challenge by providing computing services between a central networking server and end-users, organised by networking edge devices or clients [5,6]. FOG/EDGE computing outperforms cloud computing services in response time due to its distributed structure and the ability to process and store data at the end-user, near network edg devices [7]. However, to efficiently and securely serve applications anywhere, anytime, IoT must connect diverse devices and facilities using various network technologies. This underscores the importance of exploring IoT's concept and features in developing and deploying FOG/EDGE computing-based IoT [8].

There is increasing adoption of cloud technology in modern businesses, which are benefiting immensely from it. Cloud users will be able to implement many types of services that are both practical and lucrative. Using cloud computing makes it possible to concentrate on delivering high-value/profit services with a high level of quality. Using cloud computing allows for an increased focus on providing high-value/profitable services with a high degree of quality. The cloud computing model has assisted in the development of sophisticated, multi-layered information systems, which are frequently developed with a service-oriented approach [9]. Multiple tenants may deploy their applications on one or more clouds, allowing for application deployment and flexibility via shared platforms and standards [10]. Cloud business processes entail collaborative services and resources from several security realms that require dynamic integration and implementation throughout operation [11]. Security, in this context, encompasses a set of entities (individuals, services, and machines) authorized with absolute authority for authentication (a trusted principal) in a specific security processes range, orchestrating access policies to services and resources. During the authentication process, the principal must declare the credentials provided by the security authentication authority to establish its identity. A robust framework is essential for securing data and operational integrity, mitigating the risk of breaches and ensuring trustworthiness in IoT and IIoT systems. In the upcoming subsection (Section 2), we will examine the realm of authentication, specifically focusing on emerging approaches, including multi-party authentication, within the context of IoT.

2 RELATED WORK

At present, distributed applications involve increasing computerization and an equally increased amount of enthusiasm [12,13]. In order to allow for a dynamic business response, a business approach should typically be versatile for both application and method. This results in the absence of any predictability in the output sequence, even though the actual phase of execution is often "one-of-a-kind"

[14]. The applications and services used in the process are heterogeneous and can therefore be provided and managed by several non-connected organizations. Companies, for example, have many security mechanisms and policies in place to safeguard their resources from security threats, but applications that reside on the resources of different companies may operate in many various heterogeneous security realms.

Multiple parties may enter or quit a session dynamically in a multiparty context. A session authority (SA) approves or rejects the session's parties. The authentication of all session participants is supervised by a SA. The SA interacts with several session handlers (SH) in practice. The SA sends a secret session key to all session participants. Whenever a new participant enters or exits, the key is refreshed and sent using forward security techniques to the active session participants[15–17]. Therefore, entire process flows can be challenging to foresee in some situations if an organisation provides user services or resources until a transaction can be executed. Increased reliance on smart, interconnected devices will lead to trillions of sophisticated communication exchanges that could constantly jeopardise personal privacy and endanger public safety and security in all aspects of our lives. The multiple interconnections and heterogeneity in IIoT of many devices and intelligent machines establish possible vulnerabilities in cyber-physical security that can later be exploited by an adversary.

In 2016, several websites, including the New York Times, Netflix, Twitter, and SoundCloud, were declared inaccessible to users by an IoT-based distributed denial-of-service (DDoS) attack [4]. For instance, in September 2016, a major distributed denial of service (DDoS) attack using defaulted passwords on IoT devices was reported [18]. The replay attack can initiate a wormhole attack [19]. Many IoT authentication techniques, however, are presented to identify and prevent replay attacks based on the timestamp in encrypted communications [15,20], the hash code [21,22], and the IPSec protocol, which uses a message authentication code-based (MAC) anti-replay mechanism.

2.1 Threat Models in the IoT

In this section, we explore various threat models within the IoT landscape. The subsequent discussion demonstrates the robustness of the proposed framework against a spectrum of attacks, showcasing its adeptness at mitigating these threats. Specifically, the framework addresses significant security concerns, including Denial-of-Service (DoS) attacks, Man-in-the-Middle (MitM) intrusions, Replay Attacks, and Impersonation Attack . The discussion sheds light on how the proposed framework's multiparty authentication handler enhances security by thwarting potential threats and ensuring the integrity of IoT communications.

Denial-of-service (DoS) - DoS cyberattacks happen when attackers attempt to prevent legitimate users from using the IoT service. DoS attacks can be detected even if the volume of attack traffic on each link is very low. One of the primary objectives of the new framework is to defend against DoS attacks by implementing a multiparty authentication mechanism that adds an extra security layer to the network and accounts. Multiparty authentication's distributed nature deters attackers from compromising a single point, ensuring continuous operation through redundancy and shifting processes to unaffected servers. Man-in-the-Middle (MitM) - involve the unauthorized interception of communication between two users (sender, S, and receiver, R), with the intent to manipulate or delete traffic. In the context of multiparty authentication, MitM attacks may exploit vulnerabilities in the exchange of authentication credentials. Such attacks are often instigated by adversaries seeking to manipulate sensor nodes into transmitting inaccurate data. By employing distinct access layers and implementing secure protocols, including cryptographic tools like digital signatures, the proposed framework can effectively prevent unauthorized interception in multiparty authentication systems.Replay Attack - A replay attack in multiparty authentication systems involves the unauthorized retransmission of previously recorded communication, posing a security threat. To counter this, the proposed framework utilizes nonces, session identifiers, and timestamping to introduce randomness, uniqueness, and time sensitivity, thereby bolstering the system's defenses against the reuse of intercepted data. Impersonation Attack – entails an unauthorized attempt to assume the identity of a legitimate user or entity. To counter this threat, the proposed framework employs robust authentication measures, including nonces, session identifiers, and timestamping. The identity of the user can only be verified by the SAC using its super secret key.

Consequently, to address the various threat models in the IoT landscape, we introduce the integration of an advanced Inter-Cloud Trust Management Model. This model plays a crucial role in establishing and sustaining trust-based associations among diverse cloud entities, ensuring integrity and reliability in collaborative sessions with multiple parties. Challenges may arise in managing the key arrangement in the Secure Access Control (SAC) vault. Featuring dynamic trust adaptation, the model adjusts levels in real-time to address evolving security threats, yet challenges in the consistent evaluation and adaptation of trust levels require attention. The model's design incorporates trust relationships to develop cross-realm authentication protocols, but challenges may emerge in ensuring seamless multi-party sessions while upholding trust and security standards.

2.2 Inter-cloud Trust and Authentication

To understand the trust and authentication challenges in an inter-cloud environment, a practical scenario must be depicted. Clouds, geographically dispersed data-centers that are managed and maintained by service providers [23], [24]. These data-centers are running multiple applications shared by end-customers, accessing data repositories owned by principals [23]. Applications, as inter-operable cloud instances, operate through resource orchestration on a cloud federation. Individual clouds can join an inter-cloud by leveraging common authentication and authorization standards, such as common identity and resource name conventions, common addressing, common protocols for trust, presence and messaging, as well as common semantics for resource search [25,23,26]. Such framework is considered an inter-cloud trust management model [23,26]. If clouds lack inter-cloud trust models, trust providers act as mediators for integration via inter-cloud trust models controlled by them. Principles ensure interoperability through a transparent trust system, involving notifications, certifications, and the destruction of data copies, preserving control and ownership [27,28], this mechanism may be viewed as indirect trust establishment. This indirect trust establishment evolves into direct trust through positive interactions, reputation, and increased rapport [28,29] (Figure 1).

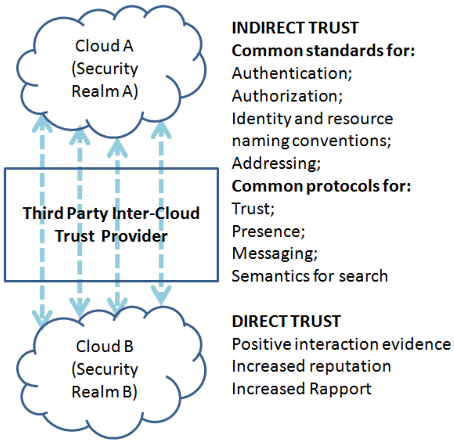


Fig. 1: Inter-cloud trust management model.

The primary challenge in implementing an authentication mechanism within a complex inter-cloud environment is efficiently managing the numerous bidirectional exchanges for approvals [30]. To address this, users require a secure system enabling single sign-on (SSO) to diverse inter-cloud services, validating their identity-related information with the requester's digital signature [30–32]. Notably, Al-Aqrabi et al. [33] introduced a multi-party authentication mechanism in a complex inter-cloud framework to establish multi-party cross-realm sessions. Their work extends beyond standard SSO by providing security credentials and introduces both indirect and direct trust elements, eliminating the time delay required for reputation and constructive interaction development. In terms of security credentials and key exchange, Dai, Q., et al. [34] introduce group credentials for key exchanges in cross-realm inter-cloud access control. New group members gain access using shared group secrets. Xu, J., et al. [35] introduce the concept of a trusted entity or authority requesting access for a new entity. For example, an IoT device or a trusted entity may request access for a new IoT device based on their established trust relationship and reputation. Initiators may request access to the new entity's active application sessions, processed by a session authority (SA). All requests are aggregated via a multi-party session handler (MPSH), creating a functional queue for the session authority.

The multiparty inter-cloud authentication system is predominantly automated, eliminating the need for initiators to manually interact with the MPSH. This enables its deployment in sensor clouds with minimal human intervention, distinguishing it from previous approaches. This study introduces an enhanced framework design suitable for authentication in sensor clouds that involve multiple parties, multiple cloud platforms, and different realms. The framework obviates the necessity for manual input from human actors. The subsequent section provides a succinct overview of sensor clouds before elaborating on the enhanced algorithm.

2.3 Sensor Clouds

Sensor clouds are cloud computing applications that use Internet-enabled sensing objects such as WSN nodes (*wireless sensor networking*), radio frequency identification (RFID) nodes, Internet-enabled devices running embedded Java beans (EJB), and mobile phones as computing devices [36]. Sensor clouds are being used in a variety of applications, including surveillance, smart cities, smart manufacturing, critical infrastructure monitoring and control, environmental monitoring, health monitoring, smart transportation, and smart logistics [36, 37]. The sensor nodes may also be actuated in industrial applications. IoT and sensors operate with typical IPv4/IPv6-based communication, allowing rapid scalability and autonomy. They may use typical IPv4/IPv6-based communication and routing protocols. Sensor nodes are easily identifiable, pervasive, and applicable in real-world processes and industrial systems [36–38, 14, 39].

Sensors generate vast data to be stored on large-scale data consolidation (DC) servers [3], referred to as "big data," and distinguished by the diversity of data, the velocity of data (*data input feeds per unit time*), the correctness of data, and the capacity of data storage (*volume*) [3, 40]. Heterogeneous sensor data are sent to a cloud-based big data analytics (BDS) servers, processed using massively parallel computing, and results are made available for visualization and various sensing/actuation applications [3, 40].

In IoTs, privacy and trust challenges involve protecting devices, data mines, computational engines, and data feeds [41]. Therefore, studies suggest that authentication and trust in sensor clouds can occur at the (1) application layer and/or (2) network layer [42]. While the physical layer contains the IoT devices, they are authenticated at the network layer by providing them with physical connections, IPv6 addresses, unique IDs, secured routing, and access keys [43, 41]. The application layer authentication is for registered users based on the underlying IoT-acquired data [43].

3 MOTIVATION AND CHALLENGES

Multiparty authentication in multi-cloud environments is challenging, especially with the increasing complexity of IoT devices. Systems with 1:1 device-to-cloud connectivity encounter added complexity, especially with the inclusion of Adaptable Wireless Sensor Networks (WSNs) which consists of diverse devices with varying accessibility and capabilities. Overcoming these challenges involves adeptly managing interactions for authentication approvals in collaborative settings and validating authentication authority across security realms with direct cross-realm connections. Establishing intermediate realms for cross-realm authentication incurs substantial costs, involving multiple credential transfer processes and intensive calls to intermediate services. A lack of links between security realms requires a conventional, lengthy process, impacting service applicability in complex e-commerce. Enforcing a group authentication protocol necessitates exchanging various security attributes, including shared secrets for the group session key, private keys, key replication, session forwarding to other clouds, and key control. However, the difficulty in defining and verifying whether a service instance is a participant in the same multiparty session poses a challenge for session users [17].

Authenticating clients for multi-party access to cloud-hosted resources presents a fundamental challenge. This process, within a multi-organizational context, involves complex definitions, policies, storage requirements, cryptographic key exchanges, and directory services [44,45]. In a multi-cloud environment, each cloud is considered a separate security realm. This introduces complexity to authentication techniques, as they must be compatible with distributed security domains across interconnected clouds. [35,17]. Clients requesting access must undergo through a series of agreements and information/key exchanges for resources on different clouds. In interconnected sensor clouds, involving non-human actors, increases the complexity of data collection, big data analytics, and remote control actuation. Trust is crucial for IoT adoption and acceptance by the industry, emphasizing the need to preserve user privacy and security. Integrating IoT with cloud computing necessitates dynamic authentication and authorization due to the dynamic nature of cloud systems. However, security and privacy risks arise in cloud technologies, particularly concerning unauthorized access to sensitive corporate information in a cloud system. Furthermore, collaboration of services from diverse security domains in business operations is challenging without defined authentication links, making safeguarding practically impossible. Therefore, new authentication methods and protocols are essential to establish trust across service instances by sharing a common secret session for all session participants.

This research aims to develop a dynamic multi-party authentication framework for cloud users, industrial sensors, and IoT services across diverse cloud systems in distinct security realms. Objectives include constructing and validating the framework model using linear temporal logic, evaluating its effectiveness through simulation, and creating a prototype authentication system. The study employs LTL for formal analysis, OPNET Modeller for performance assessment, and NuSMV Model Checker for resilience evaluation. The Session Authorization Component (SAC) regulates cloud sessions, securely storing authentication credentials and digital signatures.

4 CONTRIBUTION

This paper introduces an innovative multi-party authentication approach, enhancing secure interactions in a distributed framework encompassing IoT and multi-cloud systems. It proves beneficial in scenarios where users from diverse

IoT security domains access services under a trusted principal's supervision. Additionally, the proposed authentication system is effective and fully automated for industrial applications needing sensor data, big data analytics, or actuator control. It ensures secure multiparty session initiation, emphasizing trust link establishment between users in different security domains. Access to IoT resources is strictly controlled, requiring authorization from the IoT Cloud's Resource Principals, who have been granted ownership. This approach mitigates potential risks posed by adversaries attempting to generate, for example, an instance ID.

Ultimately, within the framework of an emerging system architecture prioritising secure scalability, we explore the challenges of handling large collections of IoT devices. Consequently, the proposed multiparty model presents two significant contributions:

– 1. SAC (Session Authority) deployment, including the SAC_{DB} (database) and SAC_{SH} (Session Handler), which ensures complete collaboration of the resource principals and, as a result, stringent initial trust relationship establishment.
2. Scalability of authentication requests is addressed by abstracting the SAC. This absraction allows hardware components such as sensors, actuators, etc., to be hidden behind the SAC and the SAC_{DB} or SAC_{SH} entities. As a result, the SAC is completely protected.

Contributions (1) and (2) combined provide a more rigorous foundation for addressing the security of the multiparty application architectures. They enable dynamic authentication while establishing a multiparty application, ensuring that only a minimal number of security credentials are converted when services are obtained.

5 DETAILS OF THE AUTHENTICATION FRAMEWORK AND ITS MODELLING

In alignment with the discussions in Section 3, emphasising the necessity of establishing trust across various layers, including the network, user, and data access layers, is essential. Additionally, there is a need for the advancement of lightweight authentication, access control, cryptography, and session encryption techniques to enhance the security of IoT layers [46, 47]. Building upon these insights, this section introduces the foundational principles for the proposed authentication framework designed for facilitating inter-cloud access to sensor clouds. Subsequent points of explanation will elucidate its operational aspects along with Figure 2.

The fundamental guidelines that shape the behavior and interactions within the proposed authentication mechanism architecture are presented here as defined principles or rules that establish the foundational framework for the authentication mechanism, as follows:

- **Communication Restrictions**: Direct communication with individual cloud members, including IoT devices or wireless sensors, is explicitly restricted. This isolation from external access ensures a secure environment. For instance, accessing an industrial sensor for data retrieval or function initiation over the network layer is strictly prohibited within the proposed authentication mechanism architecture.

- **Session Requests and Data Consolidation**: Data consolidation servers (DC) and the big data server (BDS) shall only accept session requests from different cross-realm, cross-cloud applications. These applications equipped with self-managed big data infrastructure, possess the capability to leverage cloud-specific data consolidation servers (DC) for streaming sensor/IoT data for real-time analysis. Moreover, they have established provisions to directly access the big data server (BDS) for real-time analysis tasks. Only certified and registered applications will be granted access to the DC/BDS servers.
- **Actuation Functions**: Actuation functions are solely accessible through data consolidation servers equipped with the required components, such as PLCs operating on an IPv6 network. These PLCs enable data-driven remote activation by facilitating direct data streaming from industrial sensors and executing applications for rules-based controls.
- **Access Restriction**s:
  Individual access to either data or actuation functionalities is strictly prohibited. Access is solely permitted through authorised applications, ensuring a controlled and secure environment. This measure establishes a clear boundary, reinforcing the importance of authorised and certified applications when interacting with the system.

Figure 2 illustrates the proposed authentication framework. The following detailed description explains the framework's components.

- The A_i, B_i, F, SAC, SAC_{DB}, and SAC_{SH} components operate under the inter-cloud authentication model, ensuring seamless and secure communication across diverse cloud environments. This adherence to inter-cloud authentication principles and standards enables these components to effectively verify identities and establish secure communication channels, regardless of their respective cloud providers.
- Cloud $A_i, \forall i \in \mathbb{N}$ are versatile IoT platforms designed for various purposes, encompassing applications such as transportation sensors, coastline security sensors, environmental sensors, and oceanic sensors. Clouds $B_i, \forall i \in \mathbb{N}$ represent specialised IoT ecosystems tailored for industrial applications. These clouds consist of industrial sensors utilised in process engineering, which ultimately integrate their data with Programmable Logic Controllers (PLCs) and Supervisory Control and Data Acquisition (SCADA) systems.
- Each cloud has a dedicated $DCS_i, \forall i \in \mathbb{N}$ server. The sensors communicate with the DCS servers of their respective clouds (only). $A_i/B_i \Leftrightarrow DCS_i, \forall i \in \mathbb{N}$
- The DCS servers in turn send their consolidated sensor databases to the big data server farm.
- The user entities will require access to either the data consolidation servers (*for raw data; if they want to run their own data analytics*) or the big data server farm (*for finally analysed multidimensional reports*).
- The vault holds authentication credentials and the digital signatures of the tenants of all clouds. The root keys of all the clouds are stored in the vault which has folders identifying the clouds. An active tenant will know the root key of its own cloud.

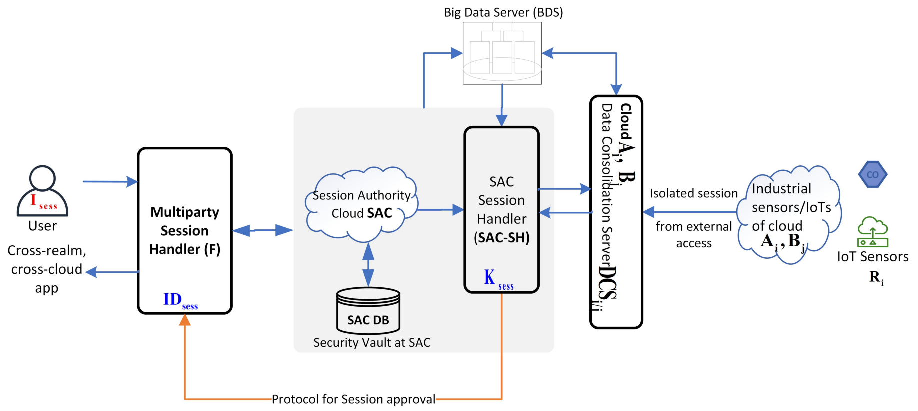


Fig. 2: Multiparty Authentication framework for Internet of Things Sensor Clouds.

This work adapts the inter-cloud authentication framework that is conceptualised and developed in [35,34,48,42]. In the authentication process, User A transmits a certificate $C(A) = \{[\mathbb{E}_{P_r(A_i)}(ID_s^A)], [\mathbb{S}_{P_r(A_i)}(ID_r^A, ID_s^A)], (ID_r^A, ID_s^A)\}$ to the multiparty handler (F), comprising a root key and an encrypted subdomain key. The subdomain key is encrypted using User A's private key ($P_r(A_i)$), including identifiers ID_r^A (cloud membership root key of user A) and ID_s^A (sub-domain membership key of user A), with a digital signature ($\mathbb{S}_{P_r(A_i)}$) signed by A private key $P_r(A_i)$. F initiates a new session ID (ID_{sess}) generation and communicates with the Session Authorization Component (SAC). SAC verifies User A's identity and grants authorization for a new session. The Session Authority Cloud (SAC) is protecting all of the servers in question, which are large-scale data servers. Each Internet of Things (IoT) cloud platform possesses its own dedicated infrastructure. The successful integration of this extensive authentication mechanism ensures a robust level of security and reliability in authenticating user identities and facilitating authorised sessions within our system. Figure 3 illustrates the different message exchanges within the proposed architecture.

- If DCS_i represent the data consolidation server for IoT device, and R_i denote the connection request $i, \forall i \in \mathbb{N}$. Then $DCS_i \rightarrow R_i$ signifies that every IoT device (i) attempts to connect (R) to its respective data consolidation server (DCS_i).
- If DCS_i represent the data consolidation server for analytics, and R_i denote the access attempt to the repositories $i, \forall i \in \mathbb{N}$. Then $DCS_i \rightarrow R_i$ signifies that the data consolidation servers (DCS_i) attempt to access the repositories of the big data analytics server(BDS).
- If A_i represent application system i, and R_i denote the connection attempt $i, \forall i \in \mathbb{N}$. Then $R_i \rightarrow A_i$ signifies that every application system (A_i) attempts to connect (R) to his DCS_i servers.
- If BDS represent the Big Data Server, DCS_i denote the i-th Data Consolidation Server, A_i represent the application, and R_i denote the application's request

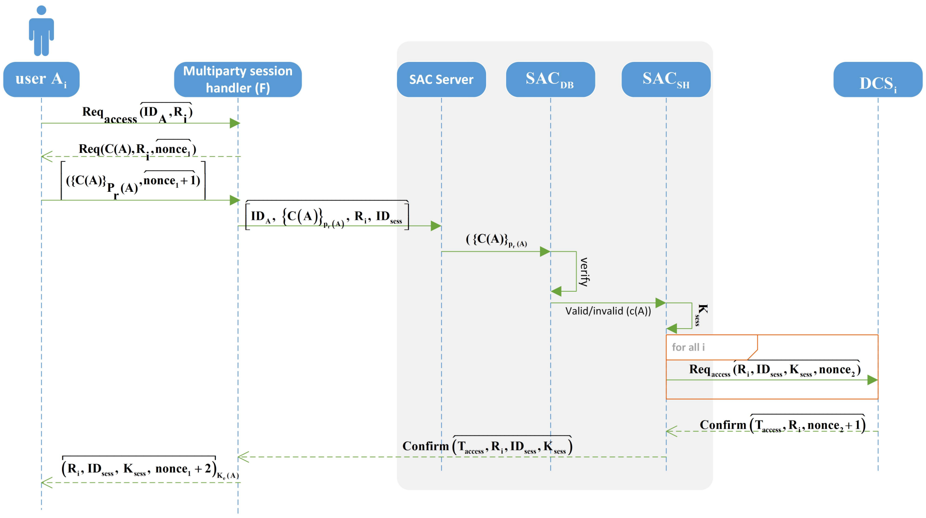


Fig. 3: The Multiparty Authentication framework interactions.

$i, \forall i \in \mathbb{N}$. Then $BDS \rightarrow R_i \leftarrow A_i \rightarrow DCS_i$ signifies that the Big Data Server (BDS) authenticates and retrieves IoT data analytics at an application's (A_i) request (R_i) from the Data Consolidation Server (DCS_i).

5.1 Linear Temporal Logic

Temporal logic, a formal language for temporal reasoning that involves understanding the sequence of events over time, is essential for formal computer system analysis, particularly in designing and validating system executions. Model checkers are integral to this process to determine if specific attributes are aligned with the system model, ensuring accuracy and adherence to given specifications. A true result suggests behavior alignment, but a counterexample reveals flaws in the design. This section deals with the authentication protocol's formal formulation and verification, leveraging temporal logic and model checkers for complete evaluation.

In the proposed system, a critical security property is established to ensure that only authorised users can request BDS access through SAC. Expressing this property in Linear Temporal Logic (LTL) formula for our security property is $\phi(P_i \rightarrow F_q)$, where P_i signifies 'User A is authorised,' and q represents 'User A granted access to the BDS.' In simpler terms, this formula asserts that for all time, if User A is authorised (P_i), then at some future time, User A will access BDS data (F_q). To illustrate, if User A gains authorization at time t_1, the formula ensures that at any subsequent time t_s, User A will eventually access BDS, reinforcing the restriction of data access to authorized users over time. For that, we employ a formal requirement specification notation based on linear temporal logic (LTL) using the temporal operators. The LTL alphabet comprises of propositional symbols (P_i, $i = 0, 1, 2, \ldots$) (we will use different capital letters to P in different contexts), booleans ($\neg, \wedge, \top, \bot,$) and temporal operators (X, Y, F, G). These elements are combined to create formulae in LTL, which are used to express requirements and conditions related to the temporal aspects of a system, and generated as follows:

$$\phi ::= P_i|\neg\phi|\phi_1 \wedge \phi_2|X\phi|Y\phi|F\phi|G\phi$$

The Boolean connectives ($\vee$, $\Rightarrow$ and $\Leftrightarrow$) will be defined in terms of $\neg$ and $\wedge$ in the usual way. In other words, the Boolean connectives $\vee$ (disjunction), $\Rightarrow$ (implication), and $\Leftrightarrow$ (biconditional) will be defined or expressed using the basic Boolean operations of negation ($\neg$) and conjunction ($\wedge$) in a standard or typical manner.

Referring to the Linear Temporal Logic (LTL) formula for our security property $\phi(P_i \rightarrow F_q)$, these Boolean connectives are commonly expressed in terms of simpler ones. Here are their typical definitions using negation and conjunction:

A user is authenticated, denoted as A. The user has a valid session key, denoted as K_{sess} to access the BDS.

1. $\neg$ (not): If $\neg$ A is true, it means the user is not authenticated.
2. $\wedge$ (and): A $\wedge$ K_{sess} is true only when both the user is authenticated (A) and has a valid session key (K_{sess}).
3. $\vee$ (or): A $\vee$ K_{sess} is true if either the user is authenticated (A) or has a valid session key ($K - sess$).
4. $\Rightarrow$ (implies): A $\Rightarrow$ K_{sess} is true if whenever the user is authenticated (A), it implies they have a valid session key (K_{sess}).
5. $\Leftrightarrow$ (if and only if): A $\Leftrightarrow$ K_{sess} is true if the user is authenticated (A) if and only if they have a valid session key (Ksession).

5.2 Semantics of LTL

Linear Temporal Logic is applied to a series of temporal states (Termed *points-in-time*, though not necessarily real-time reflections) $s_0, \ldots, s_a, \ldots$ ($a \in \mathbb{N}$). An interpretation for LTL, $I(s_a)$, at a given state s_a assigns truth values $P_i^{I(s_a)}$ to the propositional symbols P_i. A structure M is a sequence of interpretations $I(s_0), \ldots, I(s_a), \ldots$ for the sequence of states. The semantics of a LTL formula ϕ is given by a truth relationship $M, s_a \models \phi$ which means that ϕ holds at state s_a in the structure M. The relation $\models$ is defined inductively in Figure 4. The temporal operator **X** reads as *in-the-next-state*, **Y** reads as *in-the-previous-state*, **F** reads as *in-some-future-state*, and **G** reads as *in-all-future-states.* A structure M is a model of a LTL formula ϕ as follows:

5.3 Specification for the Authentication Protocol

The authentication protocol's interactions are proposed to be as follows:

$$
\begin{array}{l}
M, s_0 \models \phi \\
M, s_a \models P_i, \iff p_i^{I(s_a)} = \top \\
M, s_a \models \neg\phi, \iff M, s_a \not\models \phi \\
M, s_a \models \phi_1 \wedge \phi_2, \iff M, s_a \models \phi_1 \text{ and } M, s_a \models \phi_2 \\
M, s_a \models \mathbf{X}\phi, \iff M, s_{a+1} \models \phi \\
M, s_a \models \mathbf{Y}\phi, \iff M, s_{a-1} \models \phi \quad (a > 0) \\
M, s_a \models \mathbf{F}\phi, \iff \exists\, k \geq a \;|\!: M, s_k \models \phi \\
M, s_a \models \mathbf{G}\phi, \iff \forall k \geq a, \;\; M, s_k \models \phi
\end{array}
$$

Fig. 4: Inductive relationship of $\models$.

- User A sends a request to create a new session in order to access the IoT/Sensor clouds registered by SAC (i.e., recorded in SAC_{DB}). This is represented as ($Access\_Req_A$).
- The multiparty session handler F initiates a request for User A's keys (*certificate*), symbolized by (*Req_Cert*). This marks the initial proposition within the Linear Temporal Logic (LTL) framework, indicating the point where the certificate is requested by the multiparty session handler subsequent to a user's access request to the IoT. In LTL, this is expressed as follows:

$$
\begin{array}{l}
\phi_1 \equiv Req\_Cert \Rightarrow \mathbf{Y}(Access\_Req_A) \\
\phi_2 \equiv Req\_Cert \Rightarrow \mathbf{X}(Send\_Cert\_Handler) \\
\phi_3 \equiv Send\_Cert\_Handler \Rightarrow \mathbf{X}(Send\_SessID\_Req\_SAC) \\
\phi_4 \equiv User\_Identity\_Valid \quad \wedge \quad (Send\_SessID\_Req\_SAC) \\
\quad \Rightarrow \mathbf{X}(Send\_Req\_ID_{sess}\_Key\_Cloud)
\end{array}
$$

  This indicates that the certificate will not be requested unless the user first requests access.
- User A responds to the handler request by sending his certificate. ($C(A_i) = \{ID_r^A, ID_s^A, \{P_r(A_i)\}\}$), which includes a root key and a subdomain key that has been encrypted with user A's private key $P_r(A_i)$. $C(A_i)$ denotes the A's certificate, which is composed of ID_r^A and ID_s^A and digitally signed by $P_r(A_i)$. The user sends his or her certificate to multiparty handler F, who creates a new session ID (ID_{sess}) and sends it together with user A's request to $SAC \leftarrow (C(A), ID_{sess})$ in order to confirm user A's identity and authorize a new session.
- Using User A's public key ($P_e(A_i)$), SAC_{DB} decrypts the $C(A) \leftarrow D_{P_e(A_i)}[(C(A_i))]$. Then it may verify User A's identity by checking ID_r^A and ID_s^A. When User A validates the user identity, it is said to be legitimate ($User\_Identity\_Valid \leftarrow Verify(ID_r^A, ID_s^A)$). If User A's identity is verified, SAC generates a session key K_{sess}) and sends a request to Cloud A_i to access the IoT application. After then, SAC requests access and notifies ID_{sess} and its key of the request. Before replying to SAC, Cloud A_i records the session ID and the session key in its registry or cache *Record*(ID_{sess}, K_{sess}). SAC responds to F's session authorisation request.

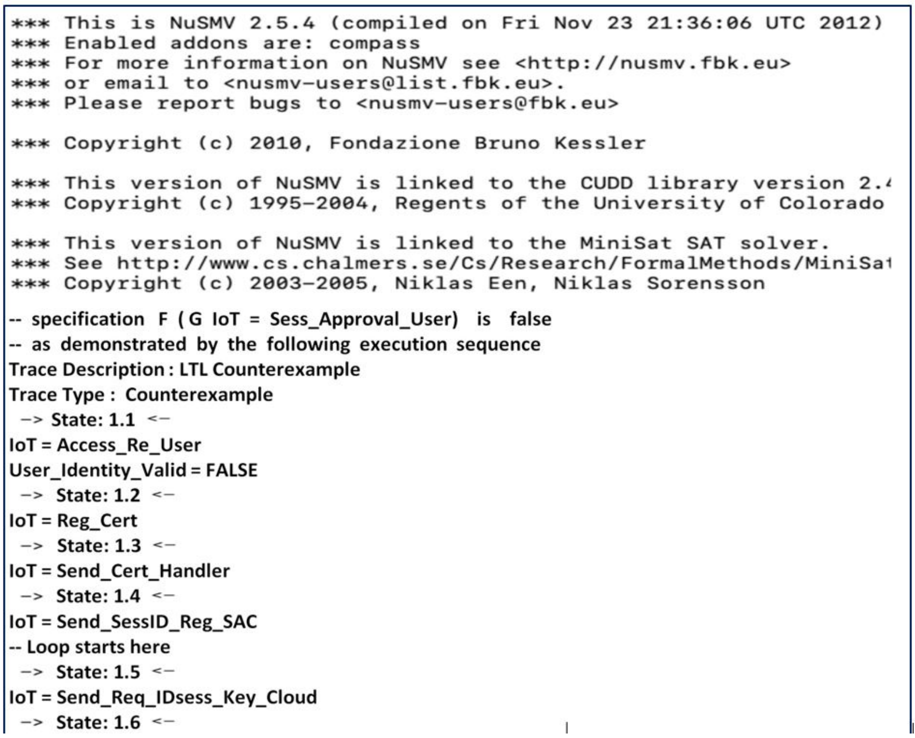
$$
\begin{array}{l}
\phi_5 \equiv Send\_Req\_IDsess\_Key\_Cloud \Rightarrow \mathbf{X}(Cloud\_Resp\_SAC) \\
\phi_6 \equiv Cloud\_Resp\_SAC \Rightarrow \mathbf{X}(Sess\_Approval\_Handler) \\
\phi_7 \equiv Sess\_Approval\_Handler \Rightarrow \mathbf{X}(Sess\_Approval\_User)
\end{array}
$$

– Following approval at F, the user receives a response for session authorisation to access the IoT application on Cloud A_i *or* B_i. The protocol begins with the user requesting access to a Cloud-based IoT application. The attributes mentioned above lead to user and session approval. This is expressed in LTL as follows:

$$\phi_{seq} \equiv \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5 \wedge \phi_6 \Rightarrow \mathbf{F}(\mathbf{G}(Sess\_Approval\_User))$$

5.4 Encoding into LTL in NuSMV

The NuSMV model checkers will be utilised to validate the proposed protocol because of their capacity to model temporal logic features and handle a large number of states to verify real-world systems. The NuSMV uses protocol features in temporary logic to provide completely automated verification [49]. In NuSVM, this is encoded as in Listing 1.



```
*** This is NuSMV 2.5.4 (compiled on Fri Nov 23 21:36:06 UTC 2012)
*** Enabled addons are: compass
*** For more information on NuSMV see <http://nusmv.fbk.eu>
*** or email to <nusmv-users@list.fbk.eu>.
*** Please report bugs to <nusmv-users@fbk.eu>

*** Copyright (c) 2010, Fondazione Bruno Kessler

*** This version of NuSMV is linked to the CUDD library version 2.
*** Copyright (c) 1995-2004, Regents of the University of Colorado

*** This version of NuSMV is linked to the MiniSat SAT solver.
*** See http://www.cs.chalmers.se/Cs/Research/FormalMethods/MiniSa
*** Copyright (c) 2003-2005, Niklas Een, Niklas Sorensson

-- specification F ( G IoT = Sess_Approval_User) is false
-- as demonstrated by the following execution sequence
Trace Description : LTL Counterexample
Trace Type : Counterexample
 -> State: 1.1 <-
IoT = Access_Re_User
User_Identity_Valid = FALSE
 -> State: 1.2 <-
IoT = Reg_Cert
 -> State: 1.3 <-
IoT = Send_Cert_Handler
 -> State: 1.4 <-
IoT = Send_SessID_Reg_SAC
-- Loop starts here
 -> State: 1.5 <-
IoT = Send_Req_IDsess_Key_Cloud
 -> State: 1.6 <-
```

Fig. 5: Session approval error counter example.

After executing the NuSMV model checkers with the provided protocol (Listing 1), the tool will return 'TRUE' indicating a successful validation and signifies the completeness of the validation process as follows:

$$\phi_{seq} \equiv \phi_1 \wedge \phi_2 \wedge \phi_3 \wedge \phi_4 \wedge \phi_5 \wedge \phi_6 \Rightarrow \mathbf{F}(\mathbf{G}(Sess\_Approval\_User)) \longleftarrow True$$

Listing 1 Proposed protocol LTL encoding syntax in NuSMV's.

```
MODULE main
VAR
IoT :
    {Access_Req_User,Req_Cert,Send_Cert_Handler,
    Send_SessID_r^Aeq_SAC,Send_Req_IDsess_Key_Cloud,
    Cloud_Resp_SAC,Sess_Approval_Handler,
    Sess_Approval_User};

User_Identity_Valid : boolean;

ASSIGN
    init (IoT) := Access_Req_User;
    init (User_Identity_Valid) := FALSE;
next(IoT) :=
    case
        IoT=Access_Req_User : Req_Cert;
        IoT=Req_Cert : Send_Cert_Handler;
        IoT=Send_Cert_Handler:Send_SessID_r^Aeq_SAC;
        IoT=Send_SessID_r^Aeq_SAC:Send_Req_IDsess_Key_Cloud;
        IoT=Send_Req_IDsess_Key_Cloud & User_Identity_Valid=TRUE: Cloud_Resp_SAC;

  IoT=Cloud_Resp_SAC : Sess_Approval_Handler;
  IoT=Sess_Approval_Handler : Sess_Approval_User;

    TRUE : IoT;
esac;

next(User_Identity_Valid) :=
   case
        IoT=Send_Req_IDsess_Key_Cloud &
   User_Identity_Valid=FALSE:TRUE;
   TRUE : User_Identity_Valid;
esac;

LTLSPEC F G(RPiIoT=Sess_Approval_User);
```

On the other hand, if the specification is not met, the NuSMV model checker will provide a counter example. For instance, if the user Identity was not validated by the SAC_{DB}, the user should not be authorized cloud access and should not have a session approval. In this situation, the NuSMV will provide a counter example rather than TRUE to the prior LTL specification. Figure 5 depicts a counter example demonstrating that specification(s) are not met.

6 RESULTS and DISCUSSIONS

For the purpose of evaluating our proposed framework, we have built a simulation environment with a reasonably large network with 1000 trusted principals accessing; a near real-world scenario.

6.1 Modelling and simulation environment

The simulation model incorporates various critical parameters and configurations to ensure a comprehensive representation of the network behavior. The "Protocol_Tasks" application relies on a task object, configured with specific phases using bespoke software in OPNET Modeller [50]. Configuration involves inputting attributes and selecting the task object bundled with the configured phases. The database application is limited to the SAC_{DB} platform, employing OPNET's default high-load task format, and its execution utilizes the profiles object. Both "Protocol_Tasks" and SAC_{DB} are configured for autonomous operation to prevent conflicts during simulation, starting with an offset between 5 to 10 seconds. Allocating time for routing protocol task execution, especially using RIPv2 as the default protocol, is justified by the network's scale.

However, the start offset of the network itself has been configured at 100 to 110 seconds to given enough time to complete the tasks of the routing protocol for large networks. Before simulations, establishing a correspondence between symbols and physical servers is crucial, retaining simplicity by preserving original symbols and server names. In the real world, meticulous OPNET model design ensures alignment between source-destination relationships and customer application phases, particularly the authentication algorithm in Protocol_Tasks. Destination preferences enable nodes to communicate only with others necessary for the algorithm, creating a recursive relationship for self-communication in OPNET. Attaching profiles to nodes guides execution, where all nodes execute "Protocol_Tasks", while the SAC_{DB} node executes both "Protocol_Tasks" and the database application. The entire multi-cloud model was created with OPNET modeller, using HP9000 Superdome 64 CPU mega modular servers. These servers are the most powerful systems available in OPNET's model library in the academic edition. Each server can host hundreds of virtual machines. The model is shown in Figure 6 and is described as follows:

1. The object A in this model comprises an Internet cloud of 1000 users, each representing a trusted principal by the SAC requesting a session each for a foreign user (member of a different security realm) on their behalf.
2. The objects F; SAC; SAC_{DB}, and SAC_{SH} are independent HP9000 Superdome servers, whereas clouds A and B are collections of four HP9000 Superdome servers each.
3. SW1 is the Internet switch for trusted principals in the inter-cloud framework, while SW2 internally connects clouds A and B to the core cloud certification authority (SAC) and its services (F, SAC_{DB}, SAC_{SH}). Both are advanced Cisco chassis-based switches with 1000 BaseX links represented by red lines, showing the number of links per connection at the center.

6.2 Session Management

In this study, we align our findings with the established concept of security realms as delineated in [35,33]. These studies identify individual business functions as distinct security realms, with participants within a session serving as authoritative responsible for facilitating the entry of new members. The initiation of a session involves a requester and an initial approver, the latter providing an instance key

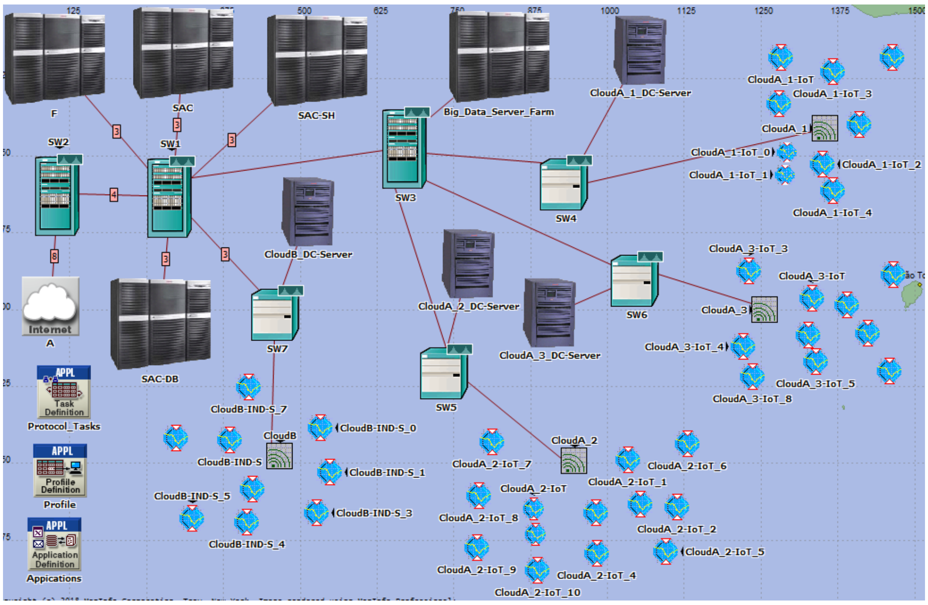


Fig. 6: The OPNET model for the proposed IoT authentication protocol framework.

I_{sess} to establish the session (i.e. the session key (K_{sess}) and the instance key I_{sess}). As an attempt to reinforce security into active sessions, Xu, J., et al. [35] introduces a directive necessitating the periodic renewal of the session key if any participant exits an ongoing session. This directive prevents reentering the session or distributing the session key without renewed consent from current session participants. However, this design lacks a mechanism for obtaining authorizations from absent principal asset owners because, in small business contexts, session participants are typically members of relevant asset owner groups. This structure helps decrease the chance of illicit key sharing [51]. On another note, cloud systems may face potential risks when the session authority issues a key based on an insider trader's instance key [52, 53]. This scenario could lead to the admission of unauthorized individuals without the awareness of absent principal asset owners, highlighting additional challenges in the design's security framework. As a consequence, our study investigates the impact of these established security procedures on the obtained results, explaining their effectiveness in maintaining the integrity and regulated access inside the security realms.

6.3 Inter and Multi cloud Interactions

Utilizing a notably intricate algorithm that doesn't consider participants of an ongoing session as the ultimate authority, the security realms were expanded into the inter-cloud framework, as outlined in [33]. This extension involves broadening the security framework to encompass interactions between multiple clouds. The sophisticated algorithm employed ensures that ongoing session participants are not treated as the final decision-makers or authorities in this extended framework. In our proposed framework, the session authority was upgraded to the Session Authority Cloud (SAC), backed by a database (SAC_{DB}) that contains real-time information about principal asset owners of cloud-hosted assets. (SAC_{DB}) can also be observed as a multi-cloud tenant registration system, controlled by a multi-cloud broker and containing information on asset ownerships.

In the model presented in [33], SAC engages with principal asset owners of destination clouds for access authorizations, despite their non-participation in the session and provision of root and sub-domain keys. The design ensures that the session key is only provided by the principal asset owners, rather than a coordinator at the asset's end. Consequently, principal asset owners can oversee approved sessions without actively participating in the session. Additionally, session requests are directed through a session handler (SAC_{SH}) to the target cloud. Conceptually, SAC_{SH} can be realized as a set of peering firewalls between the cloud broker of session requesters and the asset clouds (Clouds A_i and B_i in Figure 2), implying the existence of supplementary access control rules alongside SAC and SAC_{DB}. Building upon the same design principles, this proposed architecture has been extended to incorporate sensor clouds. As shown in Figures 2, 6, adhering to consistent design principles, the proposed architecture extends to sensor clouds, where SAC_{SH} establishes numerous peering links with clouds hosting sensors, associated DCS servers, and the BDS server. SAC is dedicated exclusively to applications requiring sensor data or BDS, and access is confined to applications appropriately registered in SAC_{DB}. The authentication protocol grants temporary approval to the session requester while concurrently forwarding resource requests to SAC_{SH} for approval from destination clouds. A representation of the protocol interactions, illustrated in Figure 2, elucidates the system's operational dynamics, how the protocol accepts requests from session owners. These requests, along with keys, are subsequently relayed to principal asset owners on sensor clouds, specifically the DC servers, as illustrated in Figure 7, which shows that the average application response time varies between 8-10 sec.

6.4 IoT Reliability

In the realm of the Internet of Things (IoT), ensuring reliable networked connectivity is crucial. Our proposed system involves creating TCP sessions, requiring rigorous testing, including stress testing for robustness, latency testing for response times, and scalability testing for handling a growing number of devices. Figure 8b shows insights into active TCP sessions exceeding 2000, averaging two session requests per trusted principal. This indicates robust utilization of the authentication protocol, triggered over 2000 times without issues. The authentication protocol efficiently operated, managing 178 million events in just seven minutes and nine seconds.

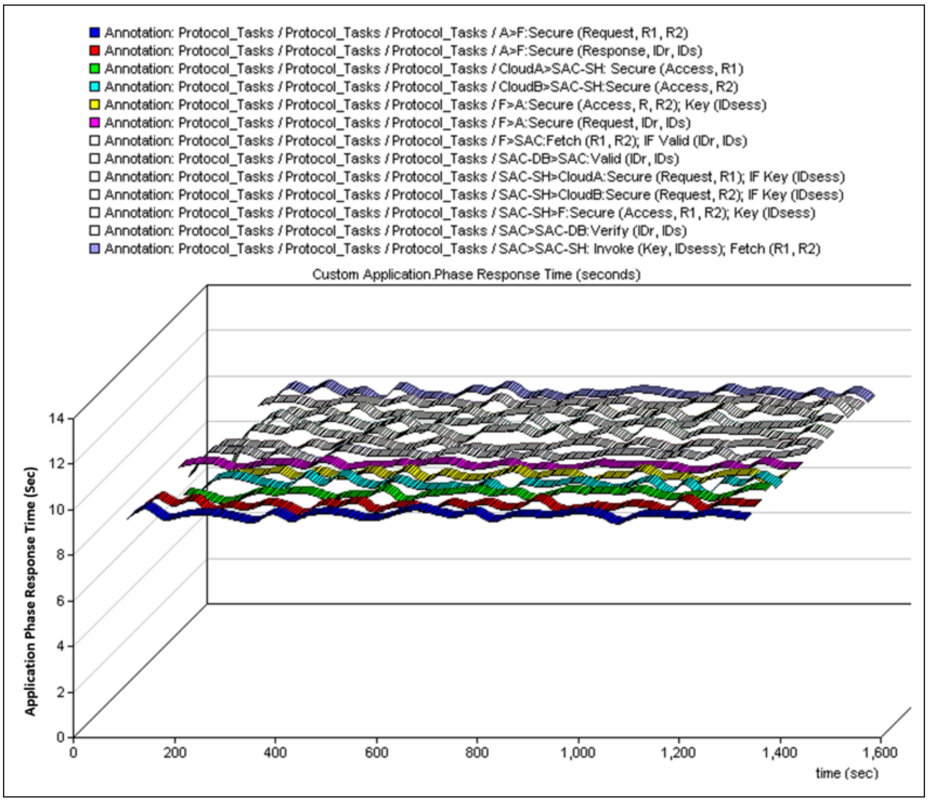
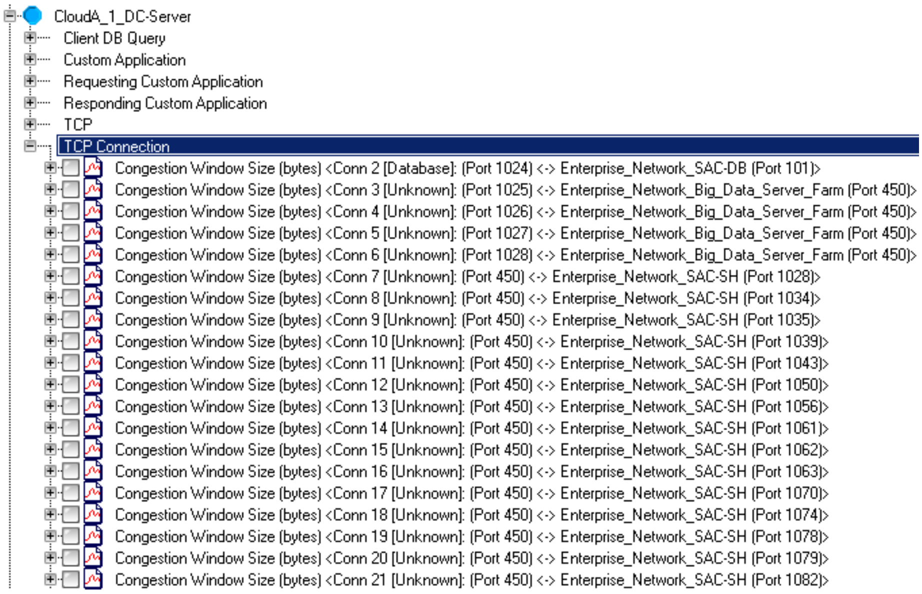
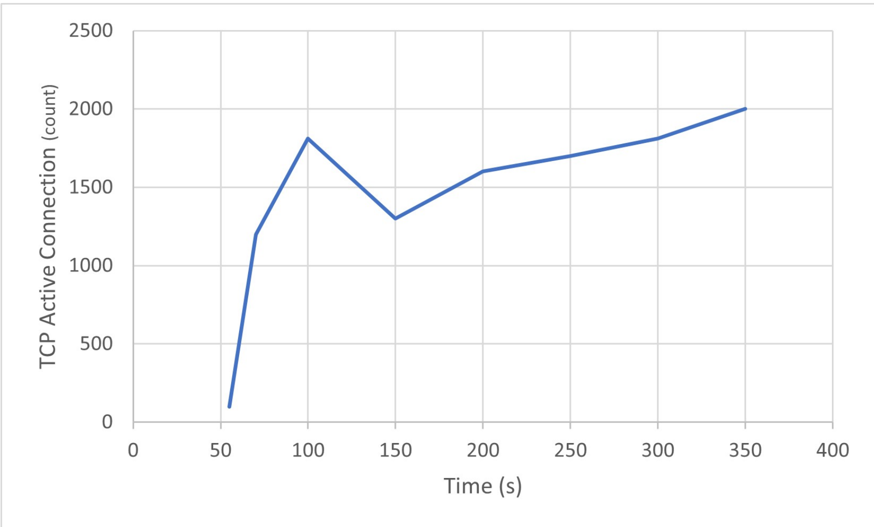


Fig. 7: Protocol_Tasks interactions and their response time.

In DCS_i servers' TCP sessions (Figure 8a), they utilize SAC_{SH}, SAC_{DB} to identify applications before granting access. Specifically, DCS_i servers reference SAC_{DB} to verify and authenticate applications during TCP sessions for DBS access. The same figure highlights that BDS servers don't directly accept queries from applications. The first SAC_{DB} session involves a one-time download of user, application system, authentication, and authorization details. Post-query, the data consolidation server continuously processes large data and session requests. Figure 8b depicts a DCS server's bidirectional TCP connections with SAC_{SH} and the BDS server. The framework designates DCS_i servers as principal owners of their sensor clouds and the BDS server cloud. However, SAC_{DB} is crucial in this architecture, serving as the major information source.



(a) TCP sessions at DCS_i.

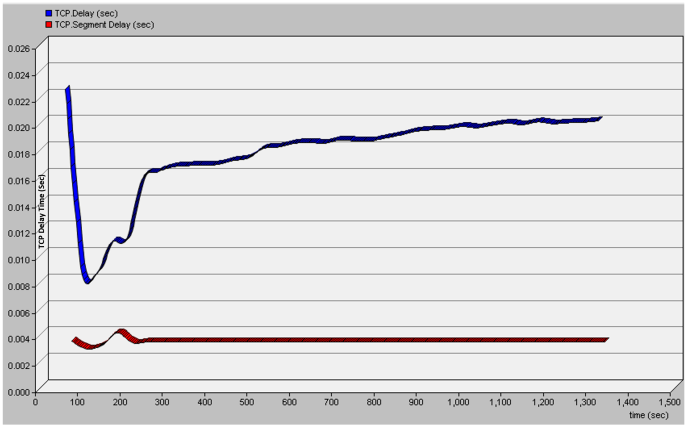


(b) Number of TCP sessions initiated by A_i at DCS_i.

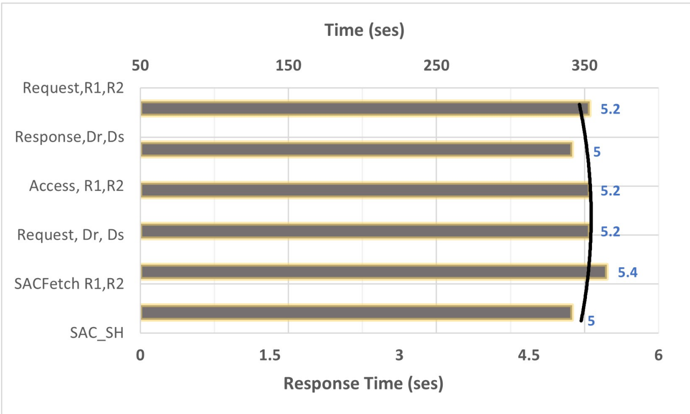
Fig. 8: TCP sessions initiated at DCS_i.

Latency testing is a crucial measure for evaluating the proposed system's reliability. Figure 9a illustrates the TCP delay characteristics of our framework over 1400 seconds. The plot reveals a gradual increase in TCP delay, primarily attributed to the initial authentication process for various connections and nodes. Notably, the delay stabilizes between 0.018-0.02 seconds from time 600 to 1400, signifying consistent and efficient performance during this period. Additionally, the

TCP segment delay per second remains stable at 0.03 seconds starting from time 250, highlighting a sustained and reliable behavior throughout the simulation. This comprehensive view of the proposed framework's temporal behavior emphasizes its stability and efficiency under varying conditions. To check the individual protocol_task response time, the TCP sessions of all sensors connecting with their respective DCS servers, the TCP sessions between the DCS servers and the BDS server, and the TCP sessions between the applications and the DCS servers, refer to Figure 9b.



(a) Protocol_tasks TCP delay time.



(b) Response times of the individual phases of the authentication protocol.

Fig. 9: Proposed system response time breakdown.

The IoTs are configured to establish multiplexed TCP sessions when the DC servers are contacting them one by one, and then respond. Further, the IoTs may respond only when the sensor data changes. Hence, there are large gaps between two consecutive streams of TCP sessions by an IoT in a sensor cloud and the beacon from the WiFi access point of a sensor cloud contacts the IoTs serially. An

IoT may respond only if any of its sensor data has changed since it has provided its last update. The IoT sessions are strictly machine-to-machine with no human intervention allowed at any level (even by the administrators). Hence, the IoT sessions will only be visible on the network monitoring panels of the DC servers. The Figure 9 depicts an intriguing finding reflected in the TCP connections data of IoTs. The initial IoT connection is with the SAC_{DB}. This is because IoTs do not respond to data requests from their DC servers without first determining if they are the correct servers to respond to. The SAC_{DB} records the allocation of a DC server to a cloud. As a result, programming an initial verification in each IoT/Wireless sensor to validate the legitimacy of the DC server requesting sensor data are simple.

As part of the analysis of protocol interactions' behavior, the complete packet network latency during the execution of Protocol_Tasks is reported in Figure 10. The graph captures a noticeable 5-millisecond spike, suggesting a transient increase in latency. However, it is noteworthy that this spike gradually stabilizes, maintaining latency levels between 3 and 3.5 milliseconds. This stability indicates a consistent and reliable performance of the proposed system over the observed duration. The transient spike, though apparent, does not compromise the overall efficiency of the protocol execution, and the subsequent stability reinforces the resilience of the system under varying conditions.

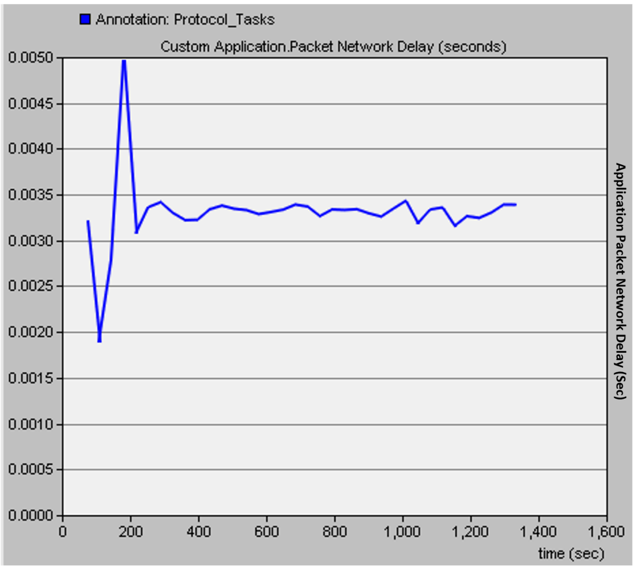


Fig. 10: Overall Packet network delay in executing the protocol.

The SAC is at the center of this architecture. Figure 11 depicts SAC's request generation timings and request-response round trip delays. Following an initial spike and drop, typical request generation durations to SAC_{DB} and SAC_{SH} were from 1.5 to 1.7 requests per second, while request-response round trips (number of requests completely processed per second) to both SAC_{DB} and SAC_{SH} were at 1 request per second. Because the protocol interactions were designed with predicted information loads, consequently, longer delays were observed.

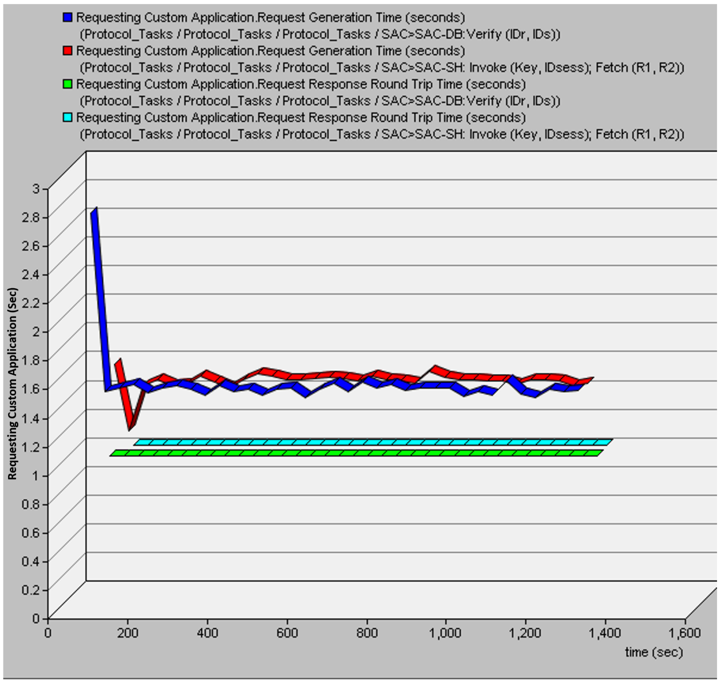


Fig. 11: Request generation times and request-response round trip delays experienced by SAC.

The overall load of requests per second and approved sessions per second is illustrated in Figure 12. The same figure translates the overall load of authentication requests and the corresponding approved sessions per second, hence, the pace at which authentication requests are generated and successfully approved, resulting in active sessions. On average, the request rate remains stable at 30 requests per second, with session initiation occurring at 10 sessions per second. This indicates an average interval of 20 seconds between the generation of a request and the initiation of a session. The observed 20-second interval suggests a steady and predictable rhythm between authentication request generation and session initiation. In an environment where the Security Access Control (SAC) lacks exclusive authority to authorize inter-cloud sessions, delays in the approval process may persist. Once sessions are authorized and established, applications can perform up to the performance limit set by the inter-cloud throughput of the cloud resource manager.

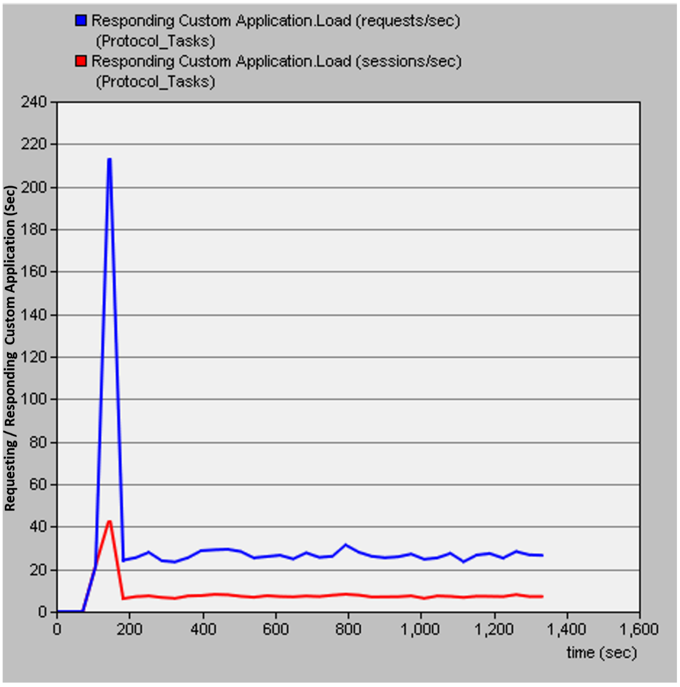
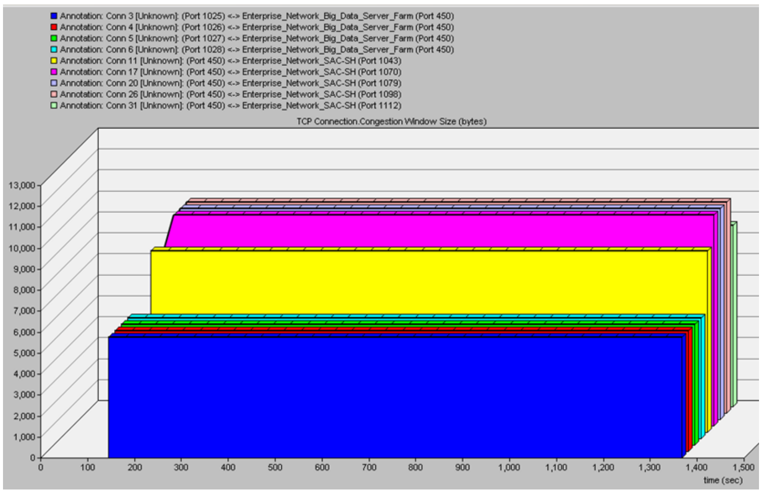
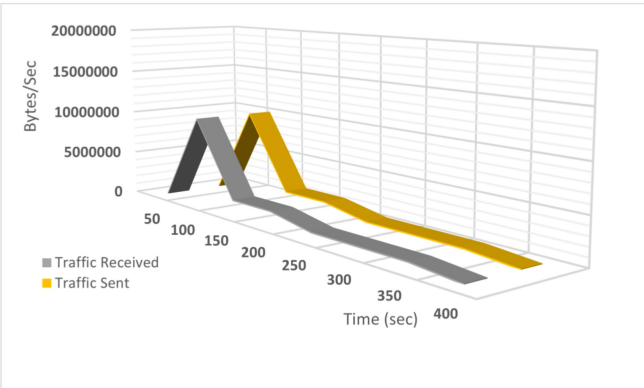


Fig. 12: Requests per second and approved sessions per second processed by the SAC.

Testing the TCP sessions/connection congestion window size is crucial for optimizing data transmission efficiency within a network. In the Transmission Control Protocol (TCP), the congestion window size governs the volume of unacknowledged data in transit, playing a pivotal role in managing network congestion and overall performance. Systematically testing and monitoring this parameter provides insights into TCP's adaptation to evolving network conditions, particularly in the context of IoT authentication. In our evaluation, we examined the DCS_i server TCP sessions/connection congestion window size, revealing distinct patterns for BDS server and SAC-SH sessions, as depicted in Figure 13a. BDS server sessions maintain remarkable stability, consistently around 4000 bytes, signifying reliable and predictable data transfer conducive to consistent authentication performance. Conversely, SAC-SH sessions exhibit dynamic profiles, with an initial increase from 7500 to 9500 bytes, followed by a subsequent decrease to 8500 bytes. These variations suggest differing behaviors in handling application requests, necessitating a closer inspection of authentication processes. Optimizing authentication in the IoT environment requires understanding these trends, underscoring the importance of tailored security measures and potential refinements in session management strategies. Additionally, when analyzing the Overall performance metrics and behaviors of the authentication Protocol Tasks on the network in terms of traffic sent and received (Figure 13b), noteworthy patterns emerge. Initially, both sent and received traffic were substantial, reaching 70,000,000 bytes. However, post the 100-time mark, a stabilization occurs, with traffic levels stabilizing at 4000 bytes. To provide a clearer perspective, this stabilization equates to approximately 3.81 MB. This observed trend aligns with the stability noted in the congestion window size for BDS server sessions, reinforcing the reliability of data transfer in the authentication process. Understanding these combined trends becomes crucial for optimizing authentication processes in the IoT environment, underscoring the importance of tailored security measures and potential refinements in session management strategies.



(a) DCS_i Server TCP sessions/connection congestion window size.



(b) Overall performance metrics and behaviours of the authentication Protocol_Tasks on the network.

Fig. 13: Proposed system overall network performance metrics.

Cloud trust brokers constitute the session authority cloud (SAC), which can grant in-principle approval for a session request from an application based on cloud root sub-domain keys. However, it requires consent from principal asset owners before allowing access to destination clouds. SAC facilitates connections among the principal owners of multiple clouds through SAC_{SH}, a mediator with its own set of policies. The proposed authentication protocol ensures security against deceptive attempts, as the session ID is never transmitted unambiguously, making it unknown to potential attackers. SAC_{DB} decrypts encrypted C(A) using User A's public key, stored securely. It then verifies ID_r^A and ID_s^A to confirm User A's identity. Both SAC_{DB} and SAC_{SH} play crucial roles, with SAC_{DB} storing essential records for SAC's knowledge, and SAC_{SH} maintaining extra network-level policy data to support additional scrutiny services.

7 CONCLUSION AND FUTURE WORK

Sensor clouds play a critical role in various applications, including control systems engineering for sensing and actuating field-level sensors/controllers. These clouds are integral to automation in industrial, commercial, and military settings, enhancing processes through large-scale analytics. The future of artificial intelligence relies heavily on these sensor clouds, but their vulnerability to hacking poses significant risks, potentially allowing unauthorized control of real-world systems such as supply chains, industrial processes, utility services, and military installations. To address this, security controls need to match the level of automation in sensor clouds and provide inherent resilience against breaches. This research extends the concept of interacting security realms to sensor clouds, ensuring complete automation. Human access is restricted, and applications are categorized within their respective security realms, managed by the SAC_{DB} cloud database supporting the session authority cloud. The proposed approach aims to serve as an effective, fully automated authentication system for applications requiring access to sensor data, big data analytics, or actuation controls. The observed response times highlight the challenges of an inter-cloud authentication protocol, signaling the need for improved links and the implementation of multiprocessing servers in authentic sensor clouds. Managing loads and response times in sensor clouds appears more intricate compared to typical application clouds with smaller databases and fewer customers.

Funding The authors did not receive support from any organization for the submitted work.

Research data policy and data availability statements All data are either included in the paper or can be found in the sources given.

Declaration

Competing Interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Compliance with Ethical Standards: This article does not include studies with human participants or animals conducted by any of the authors. The authors state that all experiments and research outlined in this manuscript adhere to ethical standards.

References

1. H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelffl e. Ccluster of european research projects on the internet of things. In *vision and challenges for realizing the Internet of Things*, page 0, European Commission, 2010.
2. D. Evans. The internet of things,howthenext evolution of the internet is changing everything. *Whitepaper, Cisco Internet Business Solutions Group (IBSG)*, 1:1–12, 2011.
3. Marcos D Assunção, Rodrigo N Calheiros, Silvia Bianchi, Marco AS Netto, and Rajkumar Buyya. Big data computing and clouds: Trends and future directions. *Journal of parallel and distributed computing*, 79:3–15, 2015.
4. Yuchen Yang, Longfei Wu, Guisheng Yin, Lijie Li, and Hongbin Zhao. A survey on security and privacy issues in internet-of-things. *IEEE Internet of Things Journal*, 4(5):1250–1258, 2017.
5. Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE internet of things journal*, 4(5):1125–1142, 2017.
6. Ahmad M Manasrah, Ala' Aldomi, and Brij B Gupta. An optimized service broker routing policy based on differential evolution algorithm in fog/cloud environment. *Cluster Computing*, 22(1):1639–1653, 2019.
7. Shanhe Yi, Cheng Li, and Qun Li. A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 workshop on mobile big data*, pages 37–42, 2015.
8. Botta Alessio, Walter De Donato, Valerio Persico, and Antonio Pescapé. On the integration of cloud computing and internet of things. *Proc. Future internet of things and cloud (FiCloud)*, pages 23–30, 2014.
9. Hussain Al-Aqrabi, Lu Liu, Richard Hill, and Nick Antonopoulos. Cloud bi: Future of business intelligence in the cloud. *Journal of Computer and System Sciences*, 81(1):85–96, 2015.
10. Richard Hill, Laurie Hirsch, Peter Lake, and Siavash Moshiri. *Guide to cloud computing: principles and practice.* Springer Science & Business Media, 2012.
11. Hussain Al-Aqrabi, Richard Hill, Phil Lane, and Hamza Aagela. Securing manufacturing intelligence for the industrial internet of things. In *Fourth International Congress on Information and Communication Technology*, pages 267–282. Springer, 2020.
12. Tariq Alsboui, Yongrui Qin, Richard Hill, and Hussain Al-Aqrabi. Towards a scalable iota tangle-based distributed intelligence approach for the internet of things. In *Science and Information Conference*, pages 487–501. Springer, 2020.
13. Tariq Alsboui, Yongrui Qin, Richard Hill, and Hussain Al-Aqrabi. Distributed intelligence in the internet of things: challenges and opportunities. *SN Computer Science*, 2(4):277, 2021.

14. Li Da Xu, Wu He, and Shancang Li. Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243, 2014.
15. Albert Wasef and Xuemin Shen. Emap: Expedite message authentication protocol for vehicular ad hoc networks. *IEEE transactions on Mobile Computing*, 12(1):78–89, 2011.
16. Mohamed Amine Ferrag, Leandros Maglaras, Abdelouahid Derhab, and Helge Janicke. Authentication schemes for smart mobile devices: Threat models, countermeasures, and open research issues. *Telecommunication Systems*, 73(2):317–348, 2020.
17. Satoshi Hada and Hiroshi Maruyama. Session authentication protocol for web services. In *Proceedings 2002 Symposium on Applications and the Internet (SAINT) Workshops*, pages 158–165. IEEE, 2002.
18. Hamid Darabian, Ali Dehghantanha, Sattar Hashemi, Sajad Homayoun, and Kim-Kwang Raymond Choo. An opcode-based technique for polymorphic internet of things malware detection. *Concurrency and Computation: Practice and Experience*, 32(6):e5173, 2020.
19. Mohamed Amine Ferrag, Leandros A Maglaras, Helge Janicke, Jianmin Jiang, and Lei Shu. Authentication protocols for internet of things: a comprehensive survey. *Security and Communication Networks*, 2017, 2017.
20. Lei Zhang, Chuanyan Hu, Qianhong Wu, Josep Domingo-Ferrer, and Bo Qin. Privacy-preserving vehicular communication authentication with hierarchical aggregation and fast response. *IEEE Transactions on Computers*, 65(8):2562–2574, 2015.
21. Hongwei Li, Rongxing Lu, Liang Zhou, Bo Yang, and Xuemin Shen. An efficient merkle-tree-based authentication scheme for smart grid. *IEEE Systems Journal*, 8(2):655–663, 2013.
22. Mian Jan, Priyadarsi Nanda, Muhammad Usman, and Xiangjian He. Pawn: a payload-based mutual authentication scheme for wireless sensor networks. *Concurrency and Computation: Practice and Experience*, 29(17):e3986, 2017.
23. David Bernstein and Deepak Vij. Intercloud security considerations. In *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, pages 537–544. IEEE, 2010.
24. Ajoze Abdulraheem Zubair and et al. A cloud computing-based modified symbiotic organisms search algorithm (ai) for optimal task scheduling. *Sensors*, 22(4):1674, 2022.
25. Lianzhong Liu and Junxiu Gao. An organization-oriented model for federated indentity management and its application. In *2008 6th IEEE International Conference on Industrial Informatics*, pages 197–202. IEEE, 2008.
26. Frank John Krautheim. *Building trust into utility cloud computing.* University of Maryland, Baltimore County, 2010.
27. Khaled M Khan and Qutaibah Malluhi. Establishing trust in cloud computing. *IT professional*, 12(5):20–27, 2010.
28. Jingwei Huang and David M Nicol. Trust mechanisms for cloud computing. *Journal of Cloud Computing: Advances, Systems and Applications*, 2(1):1–14, 2013.
29. Xiaonian Wu, Runlian Zhang, Bing Zeng, and Shengyuan Zhou. A trust evaluation model for cloud computing. *Procedia Computer Science*, 17:1170–1177, 2013.
30. Antonio Celesti, Francesco Tusa, Massimo Villari, and Antonio Puliafito. Security and cloud computing: Intercloud identity management infrastructure. In *2010 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises*, pages 263–265. IEEE, 2010.
31. Rupali Sharma and Bharti Joshi. H-ibe: Hybrid-identity based encryption approach for cloud security with outsourced revocation. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System*, pages 1192–1196. IEEE, 2016.
32. Christian Schridde, Tim Dörnemann, Ernst Juhnke, Bernd Freisleben, and Matthew Smith. An identity-based security infrastructure for cloud environments. In *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, pages 644–649. IEEE, 2010.
33. Hussain Al-Aqrabi and Richard Hill. Dynamic multiparty authentication of data analytics services within cloud environments. In *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 (HPCC/SmartCity/DSS)*, pages 742–749. IEEE, 2018.
34. Qing Dai, Xiufeng Zhao, Qiuliang Xu, and Han Jiang. A new cross-realm group password-based authenticated key exchange protocol. In *2011 Seventh International Conference on Computational Intelligence and Security*, pages 856–860. IEEE, 2011.

35. Jie Xu, Dacheng Zhang, Lu Liu, and Xianxian Li. Dynamic authentication for cross-realm soa-based business processes. *IEEE Transactions on services computing*, 5(1):20–32, 2010.
36. Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.
37. Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials*, 17(4):2347–2376, 2015.
38. Dieter Uckelmann, Mark Harrison, and Florian Michahelles. An architectural approach towards the future internet of things. In *Architecting the internet of things*, pages 1–24. Springer, 2011.
39. Roy Want, Bill N Schilit, and Scott Jenson. Enabling the internet of things. *Computer*, 48(1):28–35, 2015.
40. Bala M Balachandran and Shivika Prasad. Challenges and benefits of deploying big data analytics in the cloud for business intelligence. *Procedia Computer Science*, 112:1112–1122, 2017.
41. Qi Jing, Athanasios V Vasilakos, Jiafu Wan, Jingwei Lu, and Dechao Qiu. Security of the internet of things: perspectives and challenges. *Wireless Networks*, 20(8):2481–2501, 2014.
42. Hussain Al-Aqrabi, Anju P Johnson, Richard Hill, Phil Lane, and Lu Liu. A multi-layer security model for 5g-enabled industrial internet of things. In *International conference on smart city and informatization*, pages 279–292. Springer, 2019.
43. Maha Saadeh, Azzam Sleit, Mohammed Qatawneh, and Wesam Almobaideen. Authentication techniques for the internet of things: A survey. In *2016 cybersecurity and cyberforensics conference*, pages 28–34. IEEE, 2016.
44. Shucheng Yu, Cong Wang, Kui Ren, and Wenjing Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *2010 Proceedings IEEE INFOCOM*, pages 1–9. Ieee, 2010.
45. David W Chadwick and Kaniz Fatema. A privacy preserving authorisation system for the cloud. *Journal of Computer and System Sciences*, 78(5):1359–1373, 2012.
46. Zheng Yan, Peng Zhang, and Athanasios V Vasilakos. A survey on trust management for internet of things. *Journal of network and computer applications*, 42:120–134, 2014.
47. Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. Security and privacy challenges in industrial internet of things. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference*, pages 1–6. IEEE, 2015.
48. Hongwei Li, Yuanshun Dai, Ling Tian, and Haomiao Yang. Identity-based authentication for cloud computing. In *IEEE international conference on cloud computing*, pages 157–166. Springer, 2009.
49. Alessandro Cimatti, Edmund Clarke, Fausto Giunchiglia, and Marco Roveri. Nusmv: a new symbolic model checker. *International journal on software tools for technology transfer*, 2(4):410–425, 2000.
50. Inc. OPNET Technologies. The opnet simulator. `http://www.opnet.com`.
51. Amruta Ambre and Narendra Shekokar. Insider threat detection using log analysis and event correlation. *Procedia Computer Science*, 45:436–445, 2015.
52. Zulkefli Mohd Yusop and Jemal Abawajy. Analysis of insiders attack mitigation strategies. *Procedia-Social and Behavioral Sciences*, 129:581–591, 2014.
53. Dan Gonzales, Jeremy M Kaplan, Evan Saltzman, Zev Winkelman, and Dulani Woods. Cloud-trust—a security assessment model for infrastructure as a service (iaas) clouds. *IEEE Transactions on Cloud Computing*, 5(3):523–536, 2015.