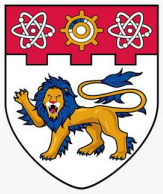
Internet of Things (IoT) security reference architecture - an ANT-centric study

Mitra, Sananda; Gondesen, Florian; Goh, Khai Hong; Lam, Kwok Yan

2020

Mitra, S., Gondesen, F., Goh, K. H., & Lam, K. Y. (2020). Internet of Things (IoT) security reference architecture - an ANT-centric study. Proceedings of the Internet of Things (IoT) Security Reference Architecture - An ANT-centric Study.

https://hdl.handle.net/10356/144422



NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

CSA SINGAPORE

INTERNET OF THINGS (IOT) SECURITY REFERENCE ARCHITECTURE AN ANT-CENTRIC STUDY

Centre for Smart Platform Infrastructure Research on Integrative Technology (SPIRIT)

Nanyang Technological University, Singapore

ABSTRACT

This document is intended to help organizations in designing secure IoT systems. It proposes a generic IoT security reference architecture that utilizes the concept of critical activities to identify appropriate security control measures for each type of node in an IoT system. The enclosed list of security control measures contains recommendations for three levels of relative strength of the mechanisms. The levels allow security architects to accommodate different requirements of an organization based on the risks and the impact of those risks analyzed for a particular application. The document also describes the steps and elaborates the prerequisites for the successful application of the proposed reference architecture for different real-life use-cases. Finally, we demonstrate the design methodology and application of the reference architecture with use cases such as Smart Lamp Posts, Smart Metering and Smart Home. This proposed technique also provides directions on how to identify and model security risks associated with different IoT systems that can help to choose the appropriate level of security controls from the reference architecture.

KEYWORDS

cybersecurity; Internet of Things (IoT); security reference architecture; security requirements; risk assessment; zero trust; activity-centric; network-centric, things-centric; smart lamp posts; smart metering; smart home

ACKNOWLEDGEMENT

The authors wish to thank all contributors to this publication, including participants in various interactive sessions; the individuals and organizations from public and private sectors; and external reviewers; whose feedback improved the technical and editorial quality of this report. The authors would also like to thank Cyber Security Agency of Singapore (CSA) for funding the research work, and Nanyang Technological University (NTU), Singapore, for operational and administrative support towards the project.

CONTRIBUTORS[1]

Florian Gondesen, Khai Hong Goh and Sananda Mitra under the supervision of Professor Kwok-Yan Lam at the Centre for Smart Platform Infrastructure Research on Integrative Technology (SPIRIT), Nanyang Technological University, Singapore.

[1] Authors are listed in alphabetical order. All authors have contributed equally to the work.

AUDIENCE AND USAGE

The audience for this publication includes:

1. Professionals who focus on application systems and business logic design can make use of this security reference to tailor security recommendations.

2. Manufacturers of IoT devices who lack access to security expertise can use the reference architecture to design the security features for devices.

3. Government agencies or business organisations who have responsibilities related to designing IoT systems.

4. IoT developers who want to design, develop and deploy secure IoT products and systems. Examples of developers include solution architects, programmers, manufacturers and system integrators etc.

5. IoT service providers who need to roll-out, configure, operate, maintain and de-commission IoT systems securely. Examples of providers include network operators, platform providers, data analysts and service delivery managers etc.

6. Users who want to procure or engage in interactions with IoT systems. For system interactions, IoT users can be either human or software agents.

The professionals using the proposed IoT security reference architecture to *generate recommendations* for their IoT systems must:

1. Develop a system architecture. The system architecture includes the placement of the devices, their connectivity and the logical data flow.

2. Perform a risk assessment to identify the impact of those risks on the system.

3. Choose the control measures referring to the list provided based on the impact of the risks to the system.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Internet of Things (IoT) has ushered an age of unprecedented opportunities for real-world applications. However, the ubiquitous connectivity of heterogeneous components over open networks and convergence of emerging technologies in IoT has introduced complexities beyond the traditional enterprise systems. The distributed and porous nature of IoT fabrics poses significant challenges for organizations in securing IoT applications and achieving resilience. The openness and scale of IoT systems also make it tough for security practitioners to establish and maintain a secure boundary as traditionally done in enterprise IT applications.

Various underlying technological elements in IoT systems bring with them their own set of cybersecurity challenges, making the IoT threat landscape extremely complicated. It is generally believed that IoT devices are lightweight and with limited resources to support adequate security measures, hence adding to the complexity of the issue. Asset visibility is also more complicated for low cost devices as they generally communicate through local gateways. Therefore, to ensure end-to-end security in IoT systems, one should concurrently consider the security issues of devices, communication channels and data processing. Given the open network configuration of IoT systems, one cannot make any assumption on the environment, and hence, the “zero trust principle” should be adopted in the design of secure IoT systems. Securing IoT systems in a zero trust environment requires strong authentication mechanisms before an entity is granted access to sensitive resources or services, or origin authenticity of data needs to be verified before they can be accepted as trusted for supporting decision making. Ensuring strong authentication and origin authenticity in a heterogeneous perimeterless infrastructure like IoT systems, requires establishing the identities of communicating parties, which in turn requires schemes and mechanisms for assigning identities, authenticating identities, and verifying the roles and privileges of the identified entities. The proposed security architecture emphasizes the importance of identity management in security protection while recognizing the challenges of enforcing identity authentication for lightweight IoT devices. For example, in situations when IoT devices are too lightweight to implement the notion of identity, behavioural analytics at the infrastructure

may be used. Nevertheless, as a highly dynamic and emerging area, a practical security architecture should be evolving with time and situations. Security designers and practitioners should closely align with the latest development in security challenges and feasible solutions and apply the most cost-effective identity management techniques for lightweight devices when they become mature.

With the aforementioned in mind, this work proposes an *Activity-Network-Things* (ANT) centric security reference architecture which is based on the three architectural perspectives in studying IoT systems namely device, internet and semantic. Existing IoT system architecture models proposed till date are mainly evolved from enterprise system architecture with adaptation to inherent features of IoT devices. As such, they typically focus on the network and device perspectives of IoT systems. In this work, we focus on the critical activities performed in different parts of an IoT system which may influence or have significant impact on the security of the entire system. The proposed security architecture provides means for system owners to determine the sensitivity of the data and the criticality of the activities that process these data. In this connection, the sensitivity of IoT data may be inherent to the nature of the data (e.g. video data of a key installation), or due to the environment within which such data are acquired (e.g. the temperature reading of a server processing classified data). Additionally, processing at edge nodes may turn out to be more sensitive due to the nature of the processing (e.g. face recognition models in camera video feed) rather than the raw data itself.

We thus describe and analyze the architectural aspects of IoT systems, and based on which, design the Security Reference Architecture from three juxtaposing views:

Activity-centric view: Represents the context of system components important to understand for end-to end security implementation and helps in identification of nodes vis-à-vis activities where sensitive IoT application data are stored and processed.

Network-centric view: Represents a communication framework of the IoT system. This view helps in the risk assessment and supplements information pertinent to the identification of critical activities.

Things-centric view: Represents the features of the physical things used in the IoT system. This view helps in understanding the inherent features of the heterogeneous things which helps to understand the capabilities in terms of security implementation. This view also supplements information pertinent to the critical nodes.

The intention of this study is to enable security professionals and stakeholders in comprehending, selecting and implementing security controls for diversified IoT applications. The following insights and guidelines are provided throughout the document, with examples in case of specific IoT applications:

1. Discussion on security challenges for the deployment of large scale IoT applications;
2. Discussion on methodologies for performing risk analysis based on standard practices;
3. Discussion on the fundamental perspectives in designing security reference architectures;
4. Identification of generic “critical activities” in IoT systems based on Activity-centric view;
5. Design of generic IoT security reference architecture in line with standards and best practices;
6. Categorization of High-Medium-Low security levels for appropriate control measures in IoT;
7. Guidelines to tailor security recommendations based on impacts identified through risk assessment;

This work is intended to be a basis for designing and planning security solutions for generic IoT systems. Depending on the management objectives of the organization and the nature of the IoT system at hand, security practitioners may choose to add or modify the proposed measures in order to incorporate customizations for their own IoT use cases and applications. The report presents sample security recommendations for specific IoT applications like Smart Lamp Posts, Smart Metering and Smart Home, to illustrate the interpretation and usage of the proposed security reference architecture in practice.

The scale of an IoT system is usually much bigger than any standard enterprise setup; it generally comprises tens of thousands of devices being deployed over a wide geographic area. Consequently, the network connectivity of IoT system is much more complicated. The fact that many IoT devices need to be deployed in open areas without direct supervision makes it difficult to enforce perimeter networks and security features in the design. Some IoT devices even have limited computing capabilities, thus prohibiting them from supporting traditional security mechanisms to enforce access control and communication security. Developing a holistic view of security requirements and applicable controls in heterogeneous IoT systems is essential for formulating a risk management strategy for design and implementation of smart IoT infrastructure. In this work we consider security requirements and controls based on the fundamental elements of IoT systems and core critical activities, to provide a secure-by-design IoT reference architecture. This architecture is flexible enough to configure the security levels and requirements commensurate with the practical risks of an IoT system, promoting secure functionality.

1.1 Problem Statement and Motivation

IoT being an emerging technology, thus far there is not enough large-scale real-world deployment experience with IoT systems to understand the security requirements and to study the security models. To deep dive into the nature of security problems in IoT systems, it is important to understand the basic characteristics of IoT systems in comparison and in contrast with traditional enterprise IT infrastructure. Based on these inherent IoT characteristics, one may analyze the issues specific to IoT systems and the security challenges thereof.

1.1.1 Characteristics of IoT Systems

IoT systems have certain characteristics[1] [2] [3] [4] quite different from typical IT systems, as follows:

- IoT devices lie at the intersection of sensor end-points and computing nodes, and thus, they interact with the physical world in a different manner compared to traditional IT devices.
- Due to operational requirements in the field, many IoT devices must comply with stringent measures of performance, reliability and resilience, quite unlike those in IT systems.

- Quite a large number of IoT devices are deployed in uncharacteristic locations. Thus, they need to be handled very different from traditional IT devices in terms of management, monitoring and servicing such as updates of software.
- Unlike devices in IT systems, IoT devices do not provide sufficient visibility into their operation, state, identity of entities they interact with, or internal software configuration and management.
- In contrast with typical IT devices, the pre-market and post-market capabilities of IoT devices are fairly limited in terms of their complex features, customizations and management.

1.1.2 Security Challenges in IoT

Given the unique characteristics of IoT systems, and compared to the typical IT setup, challenges associated with security in an IoT environment are unique too. Major security issues [2] that force one to rethink the whole cybersecurity paradigm for IoT, in contrast with IT, are as follows:

1. The complexity of an IoT system in terms of the number of devices connected and wide-ranging communication and processing requirements can make it very difficult to apply security controls.
2. IoT systems collect a wide variety of sensitive data that may have diverse security requirements.
3. Deployment in the field leaves nodes exposed to physical attacks and increases the security issues.
4. Due to the large volume of complex interactions between various heterogeneous devices, the IoT platforms, protocols, and communication gateways may allow several non-standard attack vectors.
5. IoT devices that have limited amounts of power, storage, memory or processing capability may not support complex cryptographic operations, as is required to enforce standard security controls.
6. Minor flaws in pre-market or post-market configuration of edge computing nodes can provide adversaries access to aggregated information and can act as an entry point for the entire IoT system.

7. IoT products are often not compliant with standard pre-market and post-market security standards, and stakeholders may also be uncertain about the standards and regulations to be followed.
8. Oversimplified security objectives may not cater to the complex scenario of interoperable IoT applications, and high-level reference models lacking granularity may lead to inefficient designs.

1.2 Purpose and Scope

This work is intended to be a basis for designing and planning security solutions for generic IoT systems. Albeit a promising emerging technology, there are not many successful large-scale deployments being reported worldwide, hence the scarcity of practical experience with IoT systems to understand the security requirements and to study the security models of such systems. In the IoT research community, there are three different perspectives [5, 6] to study IoT architectures: *Device-oriented*, *Internet-oriented* and *Semantic-oriented*. In our security analysis, we adopted a systematic approach to analyze the security requirements and architectural design of IoT applications from the three different perspectives. The Things-oriented perspective provides understanding of the nature of the security issues when data is captured in a distributed, open, and seemingly resource constrained environment. The Internet-oriented perspective helps to formulate the IoT security problem as a combination of fine-grained network security zones. The semantic-oriented approach is imperative as it provides insight into the criticality of the data being stored and processed. With the emerging trend of edge computing, data processing may be done anywhere between devices and the backend. As processing of data can itself be sensitive, we generalize the semantic-oriented architecture to cover processing activities. Our ***ANT-centric (Activity-Network-Things)*** view, inspired from the three perspectives, considers the criticality of data flow within an IoT application to be the basis for secure design. The proposed architecture focuses on the associated risks and is designed to ensure security in data acquisition, aggregation, processing and services.

The main purpose of this work is to propose a Security Reference Architecture, to assist in comprehending, selecting, and using appropriate security control measures for complex IoT use cases in practice.

The scope of this work includes:

1. Study the fundamental security challenges of IoT systems.
2. Identify IoT devices which are typically found in representative IoT systems.
3. Study the operating environment to recognize the threats for each of the identified IoT devices.
4. Study the security requirements of IoT applications, especially the distinguishing security characteristics.
5. Recommend the pertinent security control measures to address the identified security requirements.

1.3 IoT Security Architecture Design Principle

Internet of Things (IoT) interacts with the physical world and generate data to be analyzed and used for smart applications in a way that is very different from the traditional IT. IoT systems and devices are generally deployed in an uncontrolled environment, where the communication mostly takes place over open networks. This results in the overall IoT fabrics being extremely porous, with obscure and often unclear attack surfaces. Even though IoT systems are deployed for sensitive operations in the field, the lightweight nature of the devices poses practical constraints on performance, reliability and security. The IoT market is also less mature than the ones for traditional IT, since it is a much newer frontier based on multiple evolving technologies, namely, edge computing, big data, cloud computing and many more.

Given this situation, the IoT security architecture cannot make any assumption on the environment. Therefore, we propose to adopt the principle of “zero trust” in securing IoT systems, where the identification of protect surfaces, tracking the state and flow of data, and proposing appropriate control measures with varying levels of security are recommended to practitioners, manufacturers and other stakeholders.

1.3.1 Characteristics of the threat landscape

IoT typically consists of devices (things) used to measure or control physical properties which requires them to be deployed in the vicinity of those properties. This may entail that devices are physically out of control (or possession) of the owner or operator, for example weather sensor nodes on public ground or smart meters at the premises of a customer. Lack of physical control leads to permanent risks of devices being displaced, destroyed, or manipulated which may have adverse effects on confidentiality, integrity and availability. Furthermore, the utility of the system may be impeded by manipulating the physical properties the devices interact with. Therefore, devices cannot be trusted.

In IoT, communicating the measurement or control data typically involves public networks (the internet). Using public networks entails a permanent risk that data can be intercepted, manipulated or not transferred to the desired destination within the required time frame. Confidentiality and integrity of the data can be protected by cryptography, though some information might still be leaked by traffic patterns. Countering this by cover traffic is usually regarded as too costly, especially when public radio channels are used, and the devices' power consumption is of importance. To save power, new radio protocols are used in IoT that have not reached the maturity of the IP/TCP/TLS stack and might have undiscovered vulnerabilities. Public radio systems are especially threatened by denial of service attacks or jamming, resulting in high availability risks. Therefore, networks cannot be trusted.
In many IoT applications, the different stakeholders, owners, users, operators e.g., can be seen as adversarial parties. For example, end users of smart meters might want to avoid being billed for using electricity, while the provider might want to additionally sell the usage data. A contractor for maintenance of the smart meters might be tempted to replace smart meters more often than necessary if a different party can be billed. Therefore, stakeholders cannot be trusted.

IoT systems are increasingly relying on public cloud backends. As the cloud provider controls the machines running virtual instances, there is a risk that this power is used to access data handled in the cloud. Using this power does not necessarily require intent of the provider, the provider may also be forced by law or extortion. The provider may also cease the service for several reasons, rendering the data unavailable. Additionally, cloud infrastructure may be shared with other users, enabling attacks across the virtual machines. Therefore, cloud infrastructure also cannot be trusted.

1.3.2 Security in a Zero Trust Environment

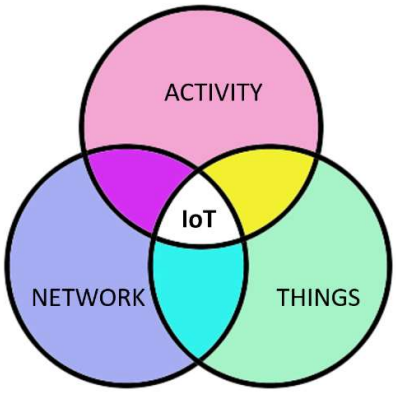
We adopt the key design principle of "securing IoT systems in a zero trust environment" to guide our proposal for the security reference architecture to address the aforesaid security issues in IoT systems. In a zero trust environment, security designers acknowledge that trust is a vulnerability. Typically, they would identify the "protect surfaces" which are made up of the network's most critical and sensitive data, assets and services in IoT systems; then identify how traffic moves across the IoT system in relation to protect surfaces; followed by putting controls in place as close to the protect surfaces as possible, creating microperimeters around them. Furthermore, critical activities and data within the protect surface do not make any security assumption about the environment and will not trust the origin and integrity of any data from outside of the protect surface unless they are explicitly authenticated and verified by strong security mechanisms by the protect surface. In this line, we propose protection of key nodes in the IoT infrastructure, as identified through our generic ANT-centric study. We take this outlook to identify the critical activities and nodes, track the state and flow of data, and propose a "zero trust" principle to secure critical activities in an IoT system. Such an approach would ensure data, network and service access to be appropriately secured. Furthermore, the threat actors who have compromised an end-node (or some end-nodes) within an IoT system will find it much harder to move laterally toward the targets with sensitive or aggregated data.

Zero trust design principle help in enforcing precise access decisions for information systems and services in the event of an asset or a part of the network being compromised. The basic tenets of zero trust [7] include:

- Authentication and authorization should be strictly enforced for all assets.
- Access to all resources should be restricted and enforced based on dynamic policies.
- Communication to and from all assets in the infrastructure should be protected regardless of location.
- Sensitive data in all states should be encrypted.
- Continuous monitoring of important activities throughout the system enforced based on dynamic policies.

1.3.3 An Activity-Network-Things-centric Study

In the IoT research community, there are three different perspectives to study IoT architectures: Things-oriented, Internet-oriented and Semantic-oriented. In our security analysis, we adopted a systematic approach to analyze the security requirements and architectural design of IoT applications from each of the three different perspectives. Finally, we present an *Activity-Network-Things* (ANT) centric risk-based study of the IoT architectures to amalgamate the three different perspectives prevalent in the cybersecurity research community.



Things-centric view in our security analysis helps to understand the nature of the security issues of heterogeneous data acquisition with the help of diverse devices not having any clear security perimeter.

Network-centric view on an architecture helps to understand the connectivity among all components and thus to identify the flow of sensitive data through the system. While IoT systems generally use the internet, local networks with IoT-specific non-standard communication protocols are also quite common.

Activity-centric analysis helps to identify critical nodes where sensitive IoT application data are stored and processed, so as to determine the critical activities of the system. Thus, we can recommend appropriate control measures based on the understanding of the acquired heterogeneous data and their transmission.

In our proposed security architecture design for IoT systems, following the zero trust principle to enforce protect surfaces of critical activities, we recommend a strong focus on the Activity aspect of the IoT infrastructure, along with Network and Things.

1.3.4 Recommended Control Mechanisms

We recommend specific control mechanisms and security levels for IoT systems in three steps, as follows:

Step 1

Critical activities and nodes are identified through the Activity-centric view of an IoT system. This is done by the practitioner after the design of the system presents the functional components. Our proposal presents a list of generic critical activities in IoT systems, as well as provides templates for specific IoT use cases. However, an experienced practitioner may choose to add or modify to our proposal.

Step 2

Control measures are mapped to each critical activity performed by the specific critical nodes. This is the job of the practitioner who has basic understanding of the role of control measures in case of each critical activity identified in Step 1. Our proposal presents a mapping of generic critical activities in an IoT system to standard control measures applicable in such cases. We also provide examples for some IoT use cases. However, an experienced security practitioner may choose to add or modify to our proposal.

Step 3

Each control measure offers three levels of security recommendations (H,M,L), as applicable. Our reference architecture was designed to be applicable to all IoT domains in general. Therefore, it contains recommendations that use the currently highest feasible security measures (H), as well as the lowest security requirements that can be considered secure to date (L). Additionally, we have the intermediate level (M) with security measures ranging between the other two levels. Our security levels are aligned with the impact levels of the FIPS 199 standard. The security practitioner may choose levels of security for the control measures based on the considerations of *reputation*, *personal/organizational interest, financial/data loss and public safety* associated with the respective critical activity. Our proposal presents some examples for such choices of security levels in case of certain IoT systems.

In summary, the proposed security reference architecture provides guidelines to specify security requirements and recommendations for a wide range of IoT applications. It also helps in developing an overall vision and strategy for control mechanisms and security compliance in targeted IoT applications.

1.4 Methodology and Models adopted

In order to achieve the research goals, our security architecture design is based on the Open System Security Model and the analysis is performed through a systematic multi-phase methodology. The phases of the research methodology, and its corresponding steps within, are as follows:

Phase 1: IoT System Architecture

Step 1. Identify distinguishing features of Internet-of-Things (IoT) systems. Analyze and compare the classical Information Technology (IT) paradigm in contrast with the Internet of Things (IoT) paradigm in order to identify distinguishing characteristics of IoT systems that will have major security implications of the security models needed by the systems. This enables us to determine the appropriate security model for IoT systems, and to guide the direction for investigating the implications of the underlying security model on the design of the IoT security architecture. The research study concluded that the Open System Security Model is needed to govern the security architecture design of IoT systems. This is because very few security assumptions can be made on the operating environment of IoT systems due to the lack of a well-defined and enforceable perimeter.

Step 2. Design a generic system architecture framework for IoT systems with security in mind. Translate the observed distinguishing characteristics into features and requirements of an appropriate system architecture design. This facilitates the study of IoT system architecture framework that addresses the characteristics of their system components, the connectivity and the processing needs. The resulting architectural features of IoT systems help to further verify the appropriateness of the Open System Security Model for IoT system security design.

Step 3. Determine the architectural features of IoT systems that are most relevant to the security-by-design of IoT applications. Apply different academic architecture models of IoT systems for assisting to identify architectural features of such systems, and to study the potential security exposure and attack surfaces of IoT systems. In this study, we applied existing IoT architecture models namely the device-oriented, internet-oriented and semantic-oriented models as our analysis tools and performed security study using different system perspectives to handle the complexity of IoT security architecture. In this connection, we devised and applied the Activity-centric, Network-centric and Things-centric views of IoT

systems to facilitate the security and risk assessment at an architectural level. This approach, which is based on the Activity-Network-Things (or ANT) centric views, allows sufficient abstraction of IoT systems for security requirement analysis.

Phase 2: IoT Security Requirements

Step 1. Identify architectural components, data and their processing that are critical to the security of IoT systems. Analysis of the security requirements of IoT systems by identifying critical activities that involve acquiring, storing, transmitting, processing and interfacing sensitive data, primarily based on the Activity-centric view of IoT systems. This is an important step in the study of risk-based security design. IoT systems involve a wide variety of devices collecting data of varying sensitivities and are connected through highly heterogeneous networks. The security requirement analysis of such systems is extremely complex, which is further exacerbated by the open nature of IoT systems. Hence, this challenging effort requires a more refined risk-based approach to address the security architecture design problem. Note that the security requirement analysis is based on the Open System Security Model where system components can make no security assumption of the environment unless security controls are explicitly implemented by the system component.

Step 2. Systematic risk assessment of IoT systems and applications based on the Activity-centric model, assisted by the Network-centric and Things-centric view of IoT infrastructures, in order to analyze the sensitivity of data, constraints on device end-points, and the overall exposure of the open network. In this study, we primarily make use of the Activity-centric model with the aim to identify critical activities which process sensitive data or aggregated data, thus requiring more stringent security control. This outcome of the Activity-centric study is enhanced with the help of the Network-centric and Things-centric studies so as to allow security architects to assess the risks that reflect on the network openness, device capabilities, and the operating environment of distributed IoT systems.

Step 3. Determination of the security requirements of IoT systems in terms of the critical activities performed by the individual nodes and entities, in conjunction with their risk exposure identified through the aforesaid ANT-centric risk assessment. The adoption of the Open System Security Model also means that the design of security controls has to be based on the Zero Trust Principle which adopts a posture where there is no implicit trust in any part

of the network and implements the necessary checks and measures. This step allows the security requirement analysis to be performed in a systematic and structured manner that identifies the critical activities to be protected that takes into consideration the security characteristics of the devices and underlying networks. The outcome of this step allows us to effectively apply risk-based security design when deciding suitable control measures for meeting the security requirements.

Phase 3: IoT Security Control Measures

Step 1. Design and develop a generic framework to identify suitable mechanisms to implement appropriate control measures to meet the security requirements at individual nodes of the IoT system. The framework provides recommendations for appropriate security control measures to address the identified security requirements. The strength of the recommended control measures is also dependent on the sensitivity of the data and the criticality of the activity being considered. Hence, each of the recommended security control measures will have three levels of strength depending on the sensitivity and criticality of the data to the owner of the IoT systems. That means, when deciding the suitability of control mechanisms, it relies on the notion of sensitivity of the data/activity which is application dependent and will only be decided by the system owner. This will be addressed in the next step.

Step 2. Instantiate the security control measures of the generic framework by customizing the security requirements and determining the notion of data sensitivity and activity criticality from the organization perspective. This step allows practitioners to choose security mechanisms depending on the sensitivity of the data associated with the respective critical activities performed by IoT nodes or components based on the realistic choices of control measures and security mechanisms based on the impact of compromise as assessed by the stakeholders of the IoT system. To facilitate the process, our methodology adopted the principles defined in FIPS 199 for guiding the system owners to determine the sensitivity level in accordance with the impact level of security incidents on the organizations. For example, whether security compromise of the data will threaten the survival of the organization, or will cause significant financial loss of the organization, or just embarrassment and inconvenience to the organization.

To summarize this study adopted the approach to first identify the distinctive features of IoT systems, and concluded that, unlike classical IT systems, an Open System Security Model is needed for governing security architecture design of IoT systems as a well-defined and enforceable perimeter is missing, and very little security assumption can thus be made on the operating environment of IoT infrastructures. Our architectural design, based on the Activity-Network-Things (or ANT) centric views, allows the security requirement analysis to be performed in a systematic and structured manner. A structured requirement analysis enables us to effectively apply risk-based security design when deciding suitable control measures for meeting the security requirements.

A security reference architecture must support a system state constituting of secure assets, secure communications and secure processing for the target IoT application. To have a granular and optimized design approach for a security reference architecture in-depth understanding of the IoT system fundamentals and the typical architectural representations is a must. The study of underlying technological aspects and the security risks and requirements associated with the different architectural elements integrated together provides a better picture to form an opinion about the security considerations. In this chapter we provide an overview of the IoT system fundamentals and show how an IoT architecture is typically represented.

2.1 IoT Systems

Internet of Things (IoT) is a paradigm interconnecting different computing devices enabling advanced services based on the existing and emerging technologies. The main purpose of IoT devices is to interact with the physical world and generate data that can then be analyzed and used for smart applications. Network connectivity in IoT not only enables remote entities to exchange data but also helps the backend in controlling physical objects out in the field [8-10]. In general, the operations of IoT systems can be categorized into the following:

Data Acquisition:
Raw data collected by the sensors accommodated in the devices till dissemination.

Data Transmission
Communication of acquired data with the help of connectivity substructure from devices to the backend.

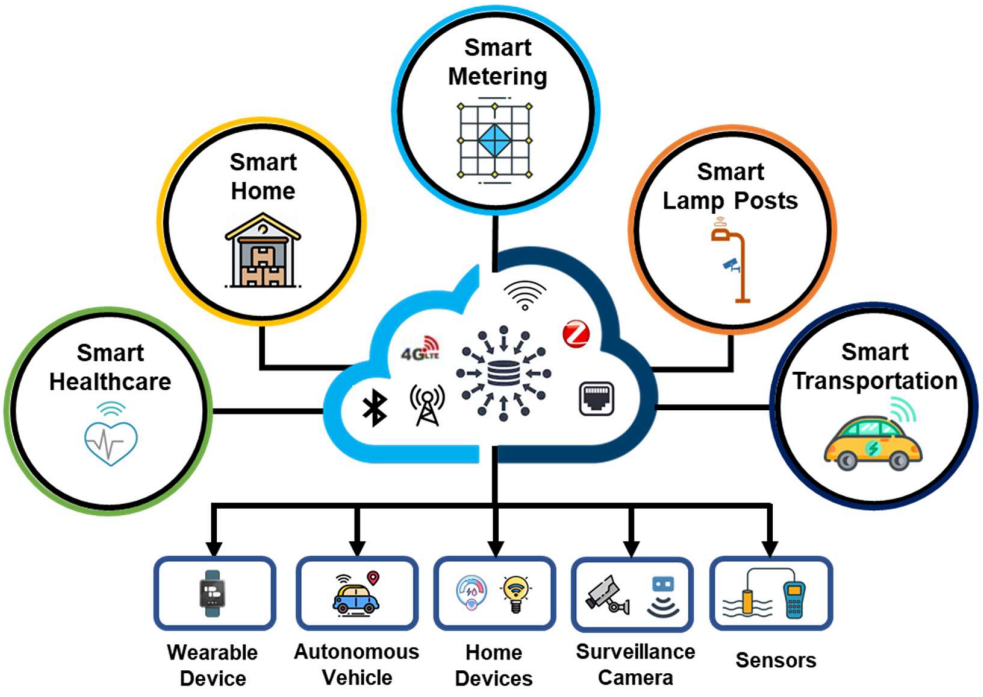
Data Analytics
Analytics on the accumulated data at the backend for example remote server or cloud service etc.

Adaptive Judgement
Real Time decisions based on the information generated from analysis.

The potential of IoT systems can be optimally utilized in large-scale applications like Smart Cities [11],[12] with the goal of improving the quality of life significantly. Smart cities make use of IoT infrastructures for solving issues ranging from traffic congestion to healthcare.

Figure 1 depicts an overview of Smart Cities including different application areas. To achieve scalability, Smart City IoT implementations must be based on an architectural design. Having an architecture allows easy integration and modification of services whenever necessary without losing functional performance.

Figure 1 Smart City IoT Example



Large-scale IoT systems promise opportunities for automation. This has motivated researchers to identify the arising challenges. Research efforts are being devoted to establishing a reference architecture so that IoT solutions can be easily implemented and be congruent with the diverse requirements.

2.2 IoT Underlying Technologies

As mentioned earlier, IoT is supported by various underlying technologies. The most important ones are discussed briefly below for clearer understanding of the enabling technologies important for this work.

Big Data for IoT

IoT systems typically gather heterogeneous data from distributed sensors. Though some kinds of sensor data can easily be structured, the overhead, for example of relational databases, needs to be considered. For performance reasons, data may be stored in semi-structured or unstructured form. Certain IoT applications even create huge amounts of unstructured data for example applications involving video surveillance. Besides storing data for later analysis, IoT systems often utilize the data gathered for real time decision making. Hence, to handle the stream of incoming data, IoT systems usually employ methods from Big data including preprocessing to reduce data and increase utility, cloud computing to gather sufficient processing power and bandwidth, and advanced machine learning techniques to make models based on the preprocessed data [3],[13]. As the processed data is used for further decisions in an IoT application data integrity is a primarily important for big data security. Data confidentiality and privacy are also quite important as big data involves an enormous amount of data storage and analytics.

Cloud Computing for IoT

Cloud computing provides on-demand access to computational resources for operations like data storage, processing and analysis [3]. IoT systems generate data which require intense computations and huge storage. Limitations of storage, energy and processing power in IoT devices prompted the use of cloud services. Cloud platforms assist IoT by providing access to the stored data or analyzed information any time and from any location and thus help to automatically control, manage and monitor devices based on those analytical results [14]. Convergence of IoT and cloud technologies comes with challenges in security and privacy of data stored/processed on shared cloud resources, interoperability of IoT systems etc. To successfully integrate IoT and cloud functionalities suitable planning regarding the design and implementation of IoT systems is required.

Edge Computing for IoT

Cloud computing plays an important role in dealing with the massive amount of data produced by sensors in the field. Moving data computation, storage and control to cloud has become a significant trend for various IoT applications. However, supporting IoT applications needing real-time services or services with shorter response time can be very challenging for cloud-based infrastructures [3],[15]. Increasing number of IoT applications deployed for real-time data processing or delay-sensitive operations has created the need for managing data nearer to the IoT devices. Nodes enabled with higher computing resources at the edge i.e. vicinity of the sensing devices can store and process raw data to reduce the traffic load as well as the response time. Multiple operations that constitute an IoT application can thus be executed at the edge. Despite the benefits of edge computing with respect to performance, there are serious concerns in terms of data security and privacy. Raw and processed data handled by the edge nodes are more vulnerable to attacks due to the perimeter-less structure [16] and the edge nodes being closer to the field in terms of deployment.

2.3 IoT Architecture

IoT architecture entails a structure which provides a high-level overview of processes conducted between all involved parties like end devices, cloud, users etc. There are several approaches that can be followed to design an IoT architecture depending on the requirements of the target application or a certain organization. Developing applications for the IoT could be a challenging task due to several reasons like complexity of networks and diverse devices, protocols, software etc [17]. The presence of a generic set of guidelines can be very useful in understanding the pros and cons of choosing the different technological elements for an IoT application. The guidelines can be best represented in the form of a reference architecture depicting the relation between the different building blocks. This section presents one of the most popular formats for IoT reference architecture.

Typically, the architecture of IoT is divided into three basic layers [6],[3]: Things layer, Network layer and Application layer. To support modelling of IoT systems providing interoperability with applications of multiple parties or service providers we consider IoT architecture to be divided into four layers: 1) Things layer, 2) Network layer, 3) Platform layer, and 4) Application layer.

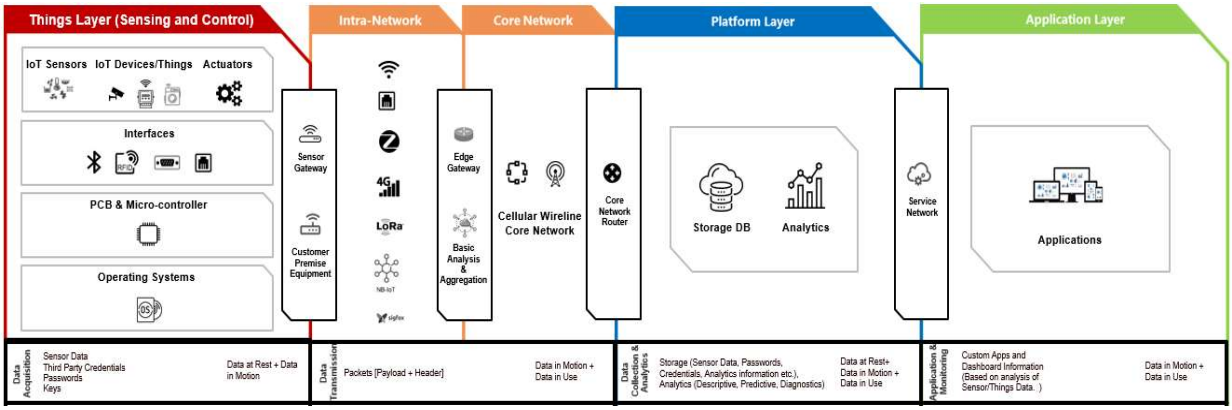
1) **Things (Sensing and Actuation / Perception Layer):** The things layer interacts with the environment and the main objectives are to collect or perceive data, hence the name. The layer also performs basic data processing if the devices are computationally capable, transmitting the processed information either for actuation or for further processing or for information display.

2) **Network Layer (Transmission Layer):** The network layer comprises of a heterogeneous and converged communication structure including short- or long-range data transmission. The network layer receives information from the Things Layer and determines how to route the data efficiently to other devices, processing platforms and applications using appropriate communication technologies.

3) **Platform Layer (Service Support Layer):** Main functions of platform layer include processing and analysis of data, providing specific information to applications/control centres and monitoring and managing end devices. For effective and consistent data services the platform layer performs necessary coordination with the application layer.

4) **Application Layer (Services Layer):** The application layer receives the processed data from the platform layer and provides required services or operations to users as well as third-party systems. This layer may also allow system administrators to manage and control the overall functionality of the IoT application. Several applications co-exist in this layer each catering to different requirements of the users of the IoT applications.

Figure 2 IoT Architecture



Each layer of the multilayer IoT architecture represents diverse functions and operations tying together all complex elements, providing easier understanding of the connectivity and interoperability. The layered Service-oriented architecture (SOA) designs like the above help in system design and provides a broad overview of the security risks but may not be effective for understanding security requirements thoroughly. It is due to the highly abstract nature of the architectures in terms of security. To achieve granular understanding of the security requirements and to provide effective guidance for accurate security controls a secure by design framework is needed. In the subsequent chapters we discuss the security risk considerations and how the proposed ANT-centric approach provides a better understanding of the step-by-step process of tailoring the security controls according to the operational environment, the functionality and the security requirements.

With the exponential increase in deployment of IoT applications all over the world, sophistication of cyberattacks directed towards them are also on the rise. IoT relies on precise perception, reliable transmission and intelligent realization of data. The main concepts to be understood before modelling IoT attacks are stated below:

Threat actors: Adversaries whose goal is to monitor or impair a target system. The treat actors can be modelled based on their capabilities, motivation and access.

***Vulnerabilities*:** Weaknesses in the system which when exposed to threat actors enable/amplify a cyberattack. Vulnerabilities can be associated with the assets or the communication framework.

Target: Entities in the system exploited by the threat actors as an entry point to gain access.

Attacks on IoT exploit the vulnerabilities in devices, network structure and even backend to disrupt, tamper or gain sensitive information. Our Things and Network-centric views allow comprehensive understanding of the risk assessment by providing different system perspectives that consider the characteristics of architectural components like the underlying networks and the distributed devices. In the next sections, we discuss common attacks targeting IoT systems in detail to provide an in-depth overview of the security risk considerations. Detailed understanding the generic risk posture helps in the efficient design of a security reference architecture as well as evaluation of its usability.

3.1 Common Attacks on IoT Systems

Attacks on IoT systems that mainly involve forging of device identity, data tampering and physical harm of sensing components [6, 18] are presented below.

a) *Node capture attacks:* The threat actor can capture and control the IoT node or device. This allows replacing or manipulating hardware or software components in the node and potentially extracting credentials or other sensitive information.

b) *Malicious code injection attacks:* In this attack, the threat actor can gain control of a node by injecting malicious code. The injected malicious code once executed can grant the threat actor access into the IoT system up to the extent of gaining full control of the IoT system.

c) *False data injection attacks:* In this attack the threat actor can inject false data to replace original measurements of the device. Transmission of false data can return erroneous feedback commands or results which further affects the effectiveness of IoT applications.

d) *Physical Damage:* In this attack hardware components are physically destroyed to make the device data unavailable. Some examples of the attack include de-packaging, destruction of flash memory etc.

e) *Sleep deprivation attacks:* In IoT, devices or nodes that are battery powered are programmed to follow a sleep cycle to preserve battery life. Sleep deprivation attacks target to break the pre-programmed sleep cycle and keep the devices awake till the time the batteries are drained, rendering the devices unavailable.

Attacks on IoT systems that have the intention to interrupt or intercept data transmission [6, 18, 19] are listed below:

a) *DoS attacks:* The threat actor renders the IoT services unavailable by overwhelming a specific target or an infrastructure with massive traffic. DoS attacks are most common in IoT systems. Common schemes for DoS attack include TearDrop, UDP flood, SYN flood etc.

b) *Routing information attacks:* Routing information attacks manipulate routing information leading to network complications like routing loops, false error messages, shortening or extending traffic routes etc. increasing end-to-end delay in IoT networks. Routing attacks mainly include spoofing, altering or replaying routing information targeting the network infrastructure and protocols.

c) *Sinkhole attacks:* A sinkhole attack is an attack on routing in which a malicious device or node advertises exceptional capabilities in terms of connectivity, computation etc. to lure neighbouring nodes to select it as the forwarding node. This node may launch DoS attacks by dropping or selectively forwarding packets.

d) *Wormhole attacks:* In a wormhole attack, two coordinated malicious nodes can exchange routing information by private links or "tunnels" to provide the false impression of one hop transmission between them, even if they are located far away from each other. Like a sinkhole attack the nodes under wormhole attack will be lured to direct more traffic through the malicious nodes to minimize forwarding hops.

e) *Sybil attacks:* In a sybil attack, a malicious node can present multiple identities and impersonate as legitimate entities in the IoT. False data sent by the malicious node can be accepted by their neighbouring node as a result of this attack. A Sybil node can also be selected as part of a routing path to transmit data leading to further attacks like jamming and DoS.

f) *Sniffing attack:* The threat actor intercepts and inspects network packets and tries to obtain sensitive information e.g. user credentials.

g) *Man-in-the-middle attack:* In this attack the attacker secretly monitors, replays or even alters the communication between two legitimate devices without their knowledge. The threat actor can pose to be either of the legitimate devices to establish cryptographic channels with both parties. As a result, the threat actor may intercept or manipulate sensitive data.

h) *Cryptanalysis attacks:* A threat actor can use intercepted ciphertext or plaintext to extract the cryptographic key. Examples of cryptanalysis attacks include ciphertext only attack, known-plaintext attack, chosen-plaintext attack, chosen-ciphertext etc.

Attacks on IoT systems that mainly affect the backend [11, 18, 19] are listed below:

a) *Unauthorized access:* Unauthorized or illegitimate access to a server, service, application etc. to gain sensitive information or control over the system.

b) *Insider attack:* An entity that is authorized to access system resources but leverages that access for malicious actions like disclosure of confidential or sensitive information.

c) *Script Injection:* Malicious scripts are executable code fragments inserted into a system to hamper system functionalities. Malicious scripts lead to the leakage or theft of sensitive data and even a complete system shut down.

A categorization of the common attacks discussed in this section, the associated risks and generic mitigation mechanism is discussed in the subsequent section.

3.2 Threat Matrix

The primary goals of risk assessment are; a) To have a clear and thorough understanding of threats and attacks to facilitate mission-relevant decision-making regarding risk level determination and risk management practices and b) To select, implement, evaluate and determine gaps in security controls for the system. Several threat analysis models exist today for efficient categorization of threats suiting the goals of an application. In this work we adopt the popular STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, Elevation of Privilege) model for categorization of the common attacks in an IoT system. The following matrix provides a basic idea how a user can identify the risks associated with the system entities and be aware of the potential impact. The user can also form a basic idea of the broad mitigation techniques which may be useful in choosing the security controls from a security reference architecture.

Table 1 Threat Matrix

Threat	Risks	Attack(s)	Target	Mitigation
Spoofing	Communication on behalf of the legitimate entity	Node capture, Identity spoofing, Routing information attack, Sinkhole, Wormhole, Sybil etc.	Devices Communication framework	Authentication mechanisms
Tampering	Physical damage, Modification of exchanged data, Reverse Engineering	Malicious code injection, False Data injection, Routing information attack, Sinkhole, Wormhole, Sybil, Man-in-the-middle etc.	Devices Data processing framework	Tamper resistant/evident package Hardware Root-of-trust Secure Boot Message Authentication Codes (MACs) Digital Signatures
Repudiation	Manipulation of data amounting to deniability	Node capture, Malicious code injection, Routing information attack, Man-in-the-middle etc.	Devices Communication framework	Digital Signatures, Certificates, PKI
Information Disclosure	Data Leaks	Node capture, Malicious code injection, Routing information attack, Sinkhole, Wormhole, Sybil Man-in-the-middle, Cryptanalysis etc.	Devices Communication framework Data processing framework	Encryption, Access Control, Intrusion Detection/Protection

Denial of Service (DoS)	Deprive the valid entities of necessary service	Node capture, Malicious code injection, Physical Damage, Sleep Deprivation, Routing information attack, Sinkhole, Wormhole, Sybil Man-in-the-middle etc.	Communication framework Devices	Redundancy Load Balancing Monitoring
Elevation of Privilege	Manipulation, Destruction, Exfiltration etc. of data Control of system entities	Node capture, Malicious code injection, Unauthorized access, Insider attack etc.	Devices Data processing framework	Authentication Access Control Software security

3.3 Risk Assessment

Risk assessment is a procedure to analyze *potential harm* that may result from vulnerabilities of the system and to quantify the degree of damage that may result from the threats exploiting such vulnerabilities. The goal of risk assessment is to prescribe practical rules or guidelines for selecting countermeasures that will bound the risks of an organization within a tolerable limit.

In practice, a system designer should perform risk assessment on the system to identify the vulnerabilities, corresponding threats and the impact of such exploits on the organization. As the threat landscape is constantly evolving in the emerging paradigm of IoT, an organization should routinely[2] perform risk assessment and implement measures accordingly. Once the risks are clearly understood, the system designer should be able to choose mitigation techniques commensurate to the impact of the risks pertaining to a specific application. The proposed security reference architecture helps in choosing the mitigation techniques as well as the levels of security controls for different impact values, but risk assessment is out of

[2] The risk assessment cycle is to be determined by the organization but is advisable to at least perform it annually.

scope for this work as it pertains to individual systems and organizations. We recommend system designers to consult the following international guidelines to assess the impacts of threats on IoT systems.

- FIPS 199 [20]: Standards for Security Categorization of Federal Information and Information Systems
- NIST Special Publication 800-63-3 [21]: Digital Identity Guidelines, Section 5.3 – Risks and Impacts

Potential Impact of the Risks

FIPS 199 defines three levels of *potential impact* on organizations or individuals in case there is a breach of security. These definitions are quite generic and the interpretation of risks under these definitions must be considered within the context of each individual organization.

Excerpt from FIPS 199, Section 3:

> The *potential impact* is **HIGH** if the loss of confidentiality, integrity, or availability could be expected to have a ***severe or catastrophic adverse effect*** on organizational operations, organizational assets, or individuals.
>
> The *potential impact* is **MODERATE** if the loss of confidentiality, integrity, or availability could be expected to have a ***serious adverse effect*** on organizational operations, organizational assets, or individuals.
>
> The *potential impact* is **LOW** if the loss of confidentiality, integrity, or availability could be expected to have a ***limited adverse effect*** on organizational operations, organizational assets, or individuals.

Risk assessment in practice may consider the level of *adverse effect* defined in FIPS 199 in conjunction with the *impact categories* defined in NIST 800-63-3 (Sec 5.3), as in the Table below.

Table 2 Impact of Risks

		Impact of Risks (ref: FIPS 199)		
		HIGH *Severe or Catastrophic Adverse Effect*	**MODERATE** *Serious Adverse Effect*	**LOW** *Limited Adverse Effect*
Impact Categories (ref: NIST 800-63-3)	harm to agency programs or public interests	major damage to organizational assets, operations, or public interests, and mission capability degradation or loss to the extent and duration that the organization is unable to perform one or more of its primary functions	significant damage to organizational assets, operations, or public interests, and mission capability degradation to the extent and duration that the organization is able to perform its primary functions with significantly reduced effectiveness	minor damage to organizational assets, operations, or public interests, and mission capability degradation to the extent and duration that the organization is able to perform its primary functions with noticeably reduced effectiveness
	unauthorized release of sensitive information	release of personal, government sensitive, or commercially sensitive information to unauthorized parties resulting in serious long-term and high impact loss of confidentiality	release of personal, government sensitive, or commercially sensitive information to unauthorized parties resulting in moderately serious loss of confidentiality	release of personal, government sensitive, or commercially sensitive information to unauthorized parties resulting in minor or inconsequential loss of confidentiality
	financial loss or agency liability	catastrophic financial loss to any	serious financial loss to any party, or a	inconsequential financial loss to any

		party, or severe agency liability, with a significantly low chance of recovery	serious agency liability, with some chance of recovery in due course of time	party, or insignificant agency liability, with a high chance of recovery
	inconvenience, distress or damage to standing or reputation	serious long-term inconvenience, distress, or damage to the standing or reputation of any party	serious short-term or limited long-term inconvenience, distress, or damage to the standing or reputation of any party	limited short-term inconvenience, distress, or embarrassment to any party
	personal safety	risk of serious injury or even a risk of death	moderate risk of minor injury or limited risk of injury requiring medical treatment	risk of minor injury not requiring medical treatment
	civil or criminal violations	civil or criminal violations that are of special importance to enforcement programs	civil or criminal violations that may be subject to enforcement efforts	civil or criminal violations of a nature that would not ordinarily be subject to enforcement efforts

To maintain a generic and flexible security model across diverse IoT applications we have classified the recommendations into three quantitative levels. As threat and vulnerability are not quantifiable values, impact is considered to be the decisive factor for choosing security controls and the choice should match the severity of a breach's impact on the business application, personal interest, public security etc. Based on the comprehensive ideas about state of data (i.e. Rest, Motion, Use) the system designers should also choose the levels of security controls. The three security levels are:

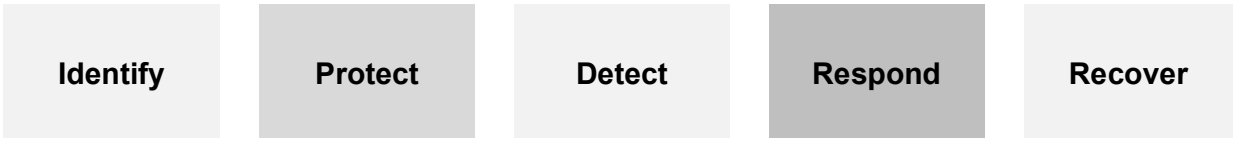
- **High**
- **Medium**
- **Low**

Each security level offers an increase in security over the preceding level (ascending order). It is advisable to maintain a single security level throughout the entire IoT system. The cryptography related recommendations for High, Medium and Low are adjusted to reach the security of 256, 192 and 128 bits respectively. The detailed levels of recommendations are discussed later in Chapter 6.

3.4 Core Functions and Association to the Security Control Measures

NIST Cybersecurity Framework [22] provides users with a framework for accessing and improving cybersecurity risk management for an organization. To manage risk of infrastructures be it IT, OT or IoT, organizations should perform proper risk assessment before making decisions about security controls or policies regarding the management of the system. With proper rigor involved in assessing the system characteristics and functionality, an organization can determine the acceptable risk level for achieving their objectives. The organizations can also quantify their risk tolerance and disseminate the information to the stakeholders. Once an organization is clear about the risk scenario, they can adjust the cybersecurity management of the system based on their target state.

Excerpts from the NIST Cybersecurity Framework state that the core functions are the highest level of abstraction included in the Framework. The five Functions namely: (a) Identify, (b) Protect, (c) Detect, (d) Response and (e) Recover were selected because they represent the five primary pillars for a successful and holistic cybersecurity program. The functions are elaborated below for the reader to understand what they individually signify.



Identify

Develop an organizational understanding to manage cybersecurity risk to systems, people, assets, data, and capabilities.

Protect

Develop and implement appropriate safeguards to ensure delivery of critical services. The Protect Function supports the ability to limit or contain the impact of a potential cybersecurity event.

Detect

The Detect Function defines the appropriate activities to identify the occurrence of a cybersecurity event. This function enables timely discovery of cybersecurity events.

Respond

The Respond Function includes appropriate activities to take action regarding a detected cybersecurity incident. This function supports the ability to contain the impact of a potential cybersecurity incident.

Recover

The Recover Function identifies appropriate activities to maintain plans for resilience and to restore any capabilities or services that were impaired due to a cybersecurity incident. It supports timely recovery to normal operations to reduce the impact from a cybersecurity incident.

The Functions should be performed concurrently and continuously to form an operational culture that addresses the dynamic cybersecurity risk. They provide a comprehensive of view the lifecycle of an organization's risk management the functions should serve as a crucial reference point.

The Functions are subdivided into groups of cybersecurity outcomes known as categories closely tied to organization needs and activities. The five core Functions and categories are as follows:

S/N	Function	Category
1.	**Identify**	a. Asset Management b. Business Environment c. Governance d. Risk Assessment e. Risk Management Strategy f. Supply Chain Risk Management
2.	**Protect**	a. Identity Management and Access Control b. Awareness and Training c. Data Security d. Information Protection Process and Procedures e. Maintenance f. Protective Technology
3.	**Detect**	a. Anomalies and Events b. Security Continuous Monitoring c. Detection Processes
4.	**Respond**	a. Response Planning b. Communications c. Analysis d. Mitigation e. Improvements
5.	**Recover**	a. Recovery Planning b. Improvements c. Communications

The five core functions are associated to the recommended control measures (refer to Table 4) based on the idea that some of categories (of each function) align totally or partially to the control measures. The Identify function is a prerequisite for using our proposed reference architecture. An organization must be aware of the impact level of security incidents and the likelihood of such incidents. The proposed reference architecture mostly provides standard cybersecurity technologies for the Protect and Detect function. A few measures also provide guidance for technologies to Respond and Recover from an incident. The organizations may break the entire system into different subsystems to employ the functions based on the need.

Given the proliferation of IoT systems in practice, security considerations should never be an afterthought. Proper planning in designing, deploying, and managing large scale IoT systems is extremely important to achieve the security goals. However, as discussed above, the traditional IT security concepts cannot be readily applied to the IoT systems[23], and one-stop generic IoT security reference designs are quite scarce[24, 25]. Application of security controls based on partially understood security requirements may add to the woes of security designers and policy makers in this space. We try to address this gap in this research work. In this section, we present an IoT security reference architecture to provide an easy and efficient way to identify the security requirements and combine the best practices and security controls for mitigating the risks posed to a generic IoT system. The design process is based on an Open System Security Model with the underlying principle of "security in a zero trust environment". We devised and applied the Activity-centric, Network-centric and Things-centric views of IoT systems to enable the security and risk assessment at an architectural level. This approach allows the security requirement analysis to be performed in a systematic and structured manner. The ANT-centric views allow the understanding of system environment, system features, typical security requirements in terms of critical activities and associated nodes in an IoT system which must be protected under any circumstances to ensure end-to-end security needs. The ANT-centric approach effectively applies risk-based security design when deciding suitable control measures for meeting the security requirements. The proposed security reference architecture provides guidelines and helps in developing an overall vision and strategy for security compliance, including options for selecting appropriate security control measures for a certain IoT system. The design is verifiable and reproducible as explained in this chapter and the subsequent use case examples.

4.1 Generic Security Requirements of IoT

Security goals can be categorized into three basic information security principles Confidentiality, Integrity, and Availability (CIA). Different IoT systems have distinctive requirements with respect to CIA property. So, a reference architecture harmonizing the heterogeneous inter-connectivity between different IoT system components is essential to guarantee end-to-end security [25]. The security reference architecture will serve the following goals:

- Anchor the security-by-design[3] discipline by providing a good development framework for project teams assuring security of various IoT applications.
- Address protection from breaches in Confidentiality, Integrity, and Availability.
- Provide effective direction for utilizing existing technologies to address pertinent IoT security challenges.

4.1.1 Things-centric Security Requirements

In our security analysis, the things-oriented view helps to understand the nature of the security issues of heterogeneous data acquisition with the help of diverse devices not having any clear boundary.

IoT devices or things perform a range of tasks like sensing, actuation, storage, pre-processing etc. with the help of ubiquitous connectivity. Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks [26] discusses the different cybersecurity risks and mitigation techniques arising from the varied device capability, functionality and interaction. The publication states the following concerns regarding the IoT devices:

1. The risks associated with the device interaction with the physical world and increased accessibility. This includes information sensing, actuation, remote access, monitoring and maintenance.
2. Heterogeneity in IoT device firmware may hamper effective software management and patching.
3. Devices may consist of hardware which cannot be repaired or customized.
4. Varied life-cycle expectations may lead to security vulnerabilities due to lack of updates.

Essential Things-centric security requirements include authentication for verifying identity, tamper resistance/evidence for ensuring data integrity, life cycle management to guarantee regular updates or device disposal and execution of adequate cryptographic operations.

[3] The Security by design approach ensures security considerations are addressed at every phase through the security lifecycle processes. Activities within these security processes focus on adding security elements that should be present in any SDLC methodologies. (CSA, Singapore)

4.1.2 Network-centric Security Requirements

A Network-centric view on an architecture helps to understand the connectivity among all components and thus to identify the flow of sensitive data through networks. While IoT generally uses the internet, networks that are under a higher level of control by the IoT system will likely also be used.

Networks can be grouped into three levels of control:

Public (the internet) The IoT system has no control over the network. Data can be intercepted, altered or transmission can be hampered. It is almost always imperative to protect the data by encryption and message integrity codes. While the internet is in principle designed to cope with failing links or nodes, failing uplinks to the internet might severely hamper the IoT system, making it advisable considering utilization of redundant uplinks.
Protected The network is a part of the IoT system's infrastructure, allowing to protect it by network access control, filtering and monitoring. Using encryption and message integrity codes as well as redundancy is advisable.
Internal / Proximity The internal/proximity network of a device that can include long range radio channels or wired buses. Compromising the network of one device is equal to compromising the device. Therefore, encryption and message integrity codes may be omitted depending on the application.

There are a variety of network protocol suites relevant to IoT, coming with varying security features or even security flaws. The required level of end-to-end security of the data must be maintained while traversing all respective networks possibly having a complicated topology. This might include switching between different protocols and even multiple stages of de- and (re)encryption, complicating the key management. The choice of the applicable security protocols is essential for the security of an IoT system. As this is highly application-dependent, the proposed security reference architecture is protocol agnostic.

4.1.3 Activity Centric Security Requirements

IoT systems cannot depend on constant system integrity of individual connected devices to ensure the ongoing integrity of the whole system. Individual devices might be compromised but the system should still function properly, if the number of compromised devices is within a predetermined threshold. It is thus important to identify areas of the system which, if compromised, will lead to negative impacts on the entire system.

For brevity, we define such areas as “critical nodes” (and activities performed on these nodes are referred to as “critical activities”) of the system and recommend special attention to the protection of the proper functioning of these nodes and activities. In general, for the security of IoT systems, we designed a framework for analysing the security requirements of IoT systems and, as a reference, identified the critical activities.

The objective of the Activity-centric security requirement analysis is to identify nodes where sensitive IoT application data are stored and processed, that is to determine the critical activities of the IoT system so that appropriate control measures can be recommended based on the understanding of the acquired heterogeneous data and the connections over which they are transmitted. To identify various sensitive activities, it is important to note the context in which the system components are used, this is because the same kind of data collected at different environments may have different significance of sensitivities to the system owner (e.g. the temperature reading of a server processing classified data may be more sensitive than temperature at a public place). On the other hand, the processing at edge nodes of less sensitive data may become more sensitive due to the nature of the processing itself; for example, image processing on the video-feed of a security camera that aims to identify “wanted people” is considered sensitive because of the recognition models (instead of the actual video feed). Hence, the processing activities on such edge nodes are also an area of consideration for the Activity-centric security requirement.

Note that the critical activities listed in this sub-section only serves as a reference and may vary from organization to organization, and from application to application. Nevertheless, the framework and the principles of identifying and using the notion of critical activities should be generic enough for the purpose of analysing security requirements of IoT systems.

The security requirements identified in this work are primarily based on the Activity-centric model while the Network-centric and Things-centric models allow further refinement of the risk assessment by providing different system perspectives that consider the characteristics of architectural components like the underlying networks and the distributed devices. We then designed and developed a generic framework to identify suitable mechanisms to implement appropriate control measures to meet the security requirements at critical nodes of the IoT system. IoT system architects may use this framework to flexibly design their own security architecture by selecting the security control measures suggested in the generic framework according to the notion of data sensitivity and activity criticality acceptable to the organization.

1. **Sensing:** Any measurement of the physical world.
    - *Data presence:* Raw sensor data.
    - *Security:* An adversary can manipulate the measured variable. If the device software does not handle unexpected values properly, this can be an attack vector. Manipulated input may also affect the system's view on the physical world so that wrong decisions are made.
    - *Privacy:* Sensors are very likely to capture PII in various forms. This includes obvious forms like video and audio streams but can also be derivatives of different sensors' data.

2. **Actuating:** Any action manipulating the physical environment.
    - *Data presence:* Control commands.
    - *Security:* The impact on security and safety depends on what kind of actuator an adversary can control. Physical security can be directly or indirectly manipulated.

3. **Preprocessing:** Any lossy transformation of data (to increase usability)
    - *Data presence:* Raw and preprocessed sensor data.
    - *Security:* Loss of raw sensor information impedes data sanity checks at later stages.
    - *Privacy:* Preprocessing can either remove PII or increase the usability of the personal information.

4. **Crypto endpoints:** Performing Encryption/Decryption or Application/Verification of message integrity mechanisms.
   - *Data presence:* Plain and encrypted data, signatures, and the cryptographic keys used.
   - *Security:* Though the availability of plaintext might suffice an adversary, keys can be extracted that might facilitate further attacks.
   - *Privacy:* Plain data may contain PII

5. **Network Transport:** Transmission of any data over any kind of network.
   - *Data presence:* Data accompanied by required metadata
   - *Security:* Transmitted data can be copied, altered, blocked, delayed, or rerouted. Adversary may obtain or modify data of multiple sources.
   - *Privacy:* Data and metadata may contain PII. Multiple non-PII data can create PII datasets.

6. **Processing:** Final processing of data which is the basis for decisions and decision making
   - *Data presence:* Analysis algorithms, data of all processing stages, analysis results, decisions
   - *Security:* Besides all stages of data, the analysis algorithms can be obtained or modified.
   - *Privacy:* Data may contain PII.

7. **Data Storage:** Storage of any data
   - *Data presence:* Application dependent
   - *Security:* Availability of previous (historic) data. Deleted data might be recoverable.
   - *Privacy:* Data may contain PII.

8. **Controlling and Configuring:** Configuring, Updating, Logging.
   - *Data presence:* Configuration data, algorithms, software, firmware.
   - *Security:* If the Command & Control service is compromised, all controlled nodes configurations/firmware can be changed to a completely different functionality. This includes functionality that still appears to be the system working as intended but in fact

leaks data or uses computational resources for tasks defined by the adversary (e.g. cryptocurrency mining, DDoS). Depending on the application, actuators can even be used for profit of the adversary (e.g. gaining physical access, re-routing physical resources).

- *Privacy:* A completely compromised system cannot hold any privacy assurance.

9. **I/O Interface:** An interface allows the system to exchange data with external (authenticated) users/systems
   - *Data presence:* Application dependent
   - *Security:* An interface is accessible from outside the system boundaries and can thus be easily attacked. Attackers might retrieve confidential data or provide bogus input data to the system. Input data must be properly sanitized. An adversary controlling the interface is able to access the data it provides and can feed bogus information to the system and external users/systems.
   - *Privacy:* Data exchanged may contain PII.

The following table summarizes the critical node types, activities and specific risks associated with the different node types in an IoT system.

Table 3 Critical Activities Performed and Associated Risks

	Activities	ID	Specific Risks
1.	Sensing and Actuating	SA	May communicate false measurement data and action based on manipulated data.
2.	Preprocessing	PP	Loss of raw data can impede sanity checks at later stage.
3.	Encrypting or Decrypting sensitive data Generating or Verifying Message Authentication Codes (MACs)	CE	Leaking of private information. Falsification of data.
4.	Transporting Data	NT	Interception and modification of data. Delay.
5.	Processing	DA	May leak or falsify data that is in the final and thus most usable form.
6.	Storing	ST	Confidentiality of raw and analyzed data may be compromised.
7.	Controlling, configuring and patching underlying nodes	CC	Integrity of all underlying nodes can be compromised.
8.	Data interfacing with users or external networks/ systems	IO	May deliver/receive sensitive/bogus information to/from illegitimate/real systems.

4.2 Security Control Measures

In this study we identified fourteen basic categories of security control measures for mitigating the identified risks specific to the critical activities which are:

1. **Entity authentication:** Entity authentication is the process by which one entity (the verifier) substantiates the identity of a second entity (the claimant). The claimant delivers evidence of its identity to the verifier to substantiate the claim.

   On an insecure channel, entity authentication can be achieved by challenge-response protocols using symmetric or public-key cryptography. To prevent replay attacks, these protocols exchange nonces as challenges. The claimant proves the ownership of the key by sending a hash of the nonce concatenated with the key (symmetric key) or sending a digital signature of the nonce (asymmetric key), which includes hashing the nonce in the first step. The relevant security property of the hash function is in both cases is the pre-image resistance. Alternatively, the verifier can send a nonce to the claimant that is encrypted with the claimant's public key. Ownership of the respective private key is proved by being able to decrypt it. The decrypted nonce can be sent back to the verifier or be used to establish a common secret. Generally mutual authentication is recommended, though in some systems unidirectional authentication might be sufficient.

   To prevent spoofing attacks, all entities conveying information with an integrity requirement should be authenticated. By default, this applies to the entities performing the activities {S, CE, CC}.

2. **Key Management:** Key management refers to secure management of cryptographic keys used for encryption. This includes dealing with the generation, exchange, storage, use, replacement and destruction of keys. Cryptography relies on full entropy key material. Each entity requires a static key stored securely on the device as well as the capability to generate random numbers with full entropy. True random number generators are often not available on simple devices. There are also typically fewer alternative sources of entropy available like the activity of users or incoming network traffic. Relying on such alternative sources is especially risky when creating the static key during deployment.

Security protocols generally establish symmetric session keys. We recommended only to use forward secure protocols as in almost all cases sensitive information does not become public in the near future. If the key establishment protocol negotiates security relevant parameters like key lengths or cipher suites, it is necessary to make sure that they cannot be downgraded by a man-in-the-middle attack.

Key management is a prerequisite to perform the activity {CE}.

3. **Symmetric Key Encryption:** In symmetric key encryption the same key is used to encrypt and decrypt the message. The security of symmetric key encryption is based on the difficulty to randomly guess or brute force the corresponding key. For simple sensor nodes that send short messages, developers might be tempted to use stream ciphers for efficiency. The security of stream ciphers relies on the key stream, which may never use the same seed. It also needs to be considered that the size of messages may leak information. Although block ciphers pad data to a multiple of the block size, some information may still be leaked by the number of transmitted blocks.

   Entities performing the activity {CE} typically encrypt or decrypt messages sent over untrusted networks to ensure confidentiality. Symmetric key encryption should also be used when storing data {ST} with a confidentiality requirement. Storage encryption helps against information disclosure when the storage is physically acquired by an adversary. Destruction of storage encryption keys is a suitable additional measure to securely delete data on Solid-State Drives (SSDs).

4. **System Integrity/Hardware Security:** Assurance of correctness and reliability of hardware and software. Especially devices deployed in the field cannot rely on a correct and secure implementation of all interfaces, as the device might be compromised by physical means. This can be mitigated by system integrity mechanisms based on a secure boot process.

   System integrity is important on systems that have access to data with confidentiality or integrity requirement activities {S, PP, CE, DA, ST} as a compromised system could leak or modify data. On nodes processing data {PP, DA}, system integrity also supports the integrity of the processing and thus the integrity of the resulting data.

5. **Physical Security:** Keeping adversaries from manipulating devices by physical means. Tampering can be impeded by appropriate physical security measures. Those should be applied on entities interacting with the physical world {S} as they are often exposed in the field and on entities performing key activities {DA, ST}.

6. **Access Control:** Access control is a selective restriction of access to computational resources or data. Though there are different access control models that have different security implications, security mainly depends on appropriate policies that limit all entities to only access resources required to perform their intended task. Access rights need to be controlled at system boundaries and essential nodes {S, NT, DA, IO}.

7. **User Authentication:** A user acts as a claimant in need to verify own credentials to the system. Though users could be technically seen as entities, the limited key storage capabilities of humans and the high risk of leakage suggests to additionally employ ownership or inherence factors.

   User authentication is required for interfaces {IO} that provide non-public data or allow user input.

8. **Intrusion Detection and Prevention:** Observing network traffic and infrastructural entities to inspect and understand malicious activities. The network traffic can be observed at entities performing network transport {NT} as routers or switches. Entities responsible for data transport as well as entities processing data need to be monitored {PP, DA, ST}.

9. **Redundancy:** Resources considered in excess to handle potential failures (due to malicious activities). As entities interacting with the physical world {S} are typically exposed to physical threats in the field, a sufficient number of nodes need to be deployed to ensure normal operation when a certain number of nodes fail. Network infrastructure {NT} should also have sufficient bandwidth and redundant paths to be able to cope with a certain number of congested or broken links. As storage {ST} may fail, redundant disk arrays are a standard practice. Replicating data on multiple locations does not only help against data loss due to disasters but may also support the overall system availability.

10. **Data Integrity / Message Authentication:** Ensuring the consistency of data during all stages preventing it to be manipulated by an adversary or changed by an error. Entities performing the activity {CE} typically apply integrity protection to messages sent over untrusted networks and verify received messages to ensure data integrity. Integrity protection is especially important for command and control channels of the system as manipulation of commands or updates may compromise the whole system {CC}.

11. **Data Sanity Checks:** Any inputs to a system need to be sanitized. This is not limited to IO interfaces, also sensor data needs to be carefully checked for sanity. Sensors can fail and adversaries might be able to manipulate the physical quantities measured. As sensor data are used for decisions or directly for control loops, feeding incorrect input can lead to adverse effects without compromising the system itself. When control systems are involved even adverse physical effects are possible.

    Data sanity checks should be performed prior to any preprocessing or data analysis as it might be impeded when raw data are discarded. Incoming data through IO interface should also be sanitized {PP, DA, IO}.

12. **Configuration Management:** Managing and administering the security of functional and physical systems associated with the environments of operation. This is realized by the command and control infrastructure {CC}.

13. **Life Cycle Management:** Managing the life cycle of entities in an IoT application like deployment, penetration testing, vulnerability disclosure, secure disposal etc. This is important for any component an IoT system. Special care must be taken for devices deployed in the field as they may be hard to reach for replacement or disposal. The devices might have moved or have become difficult to identify. Therefore, devices should not only be assigned digital identifiers and keys but also physical identifiers. Identities and deployment status should also be kept in a database.

4.3 Security Evaluation

The security of an IoT system relies on appropriate control measures that are securely implemented. This generic security reference architecture aims to help selecting control measures but cannot guarantee that these will be appropriate for a specific system with specialized security needs. The reference architecture also does not cover secure implementation of the control measures. Security controls need to be implemented carefully as even a correctly implemented algorithm may leak information by side channels. Especially for cryptosystems it is advisable to rely on well-established implementations instead of creating them from scratch.

When developing security relevant components, they should be thoroughly tested, preferably by a third party. The **Common Criteria**[4] provide a framework for security evaluation. The security requirements of the component or system to evaluate, the **Target of Evaluation (TOE),** are defined in a **Protection Profile (PP)** or a **Security Target (ST)**, which may be based on a PP. An ST is individually defined for a TOE and requires Security Assurance Requirements (SAR) to be defined accordingly, by which it will be evaluated. Typically packages of SARs are used, the **Evaluation Assurance Levels (EAL)** that have specific ascending evaluation requirements. An EAL alone is not very meaningful as the ST needs to be carefully regarded. Protection profiles contain security and assurance requirements for a certain class of devices or components. Therefore, it is generally preferable to evaluate a TOE to a PP.

HIGH	MEDIUM	LOW
For high security we strongly recommend using components evaluated to their classes' PP, or if no PP available to EAL 4.	For medium security we recommend using components evaluated to their classes' PP, or alternatively to EAL 3.	For low security, evaluation by CC may seem too costly. Instead of completely skipping evaluation, security certification schemes for consumer grade IoT devices that are currently under development, may be considered.

Information about Singapore's Cybersecurity Labelling Scheme can be found here:
https://www.csa.gov.sg/programmes/cybersecurity-labelling

Products evaluated under Singapore CC scheme can be found here:
https://www.csa.gov.sg/programmes/csa-common-criteria/cc-product-list

[4] https://www.us-cert.gov/bsi/articles/best-practices/requirements-engineering/the-common-criteria

4.4 Assigning Security Control Measures to Critical Activities

Before recommending the security control measures, the relation of the security requirements with the critical activities must be understood. The following table is a mapping of the identified control measures and critical activities which can provide guidance for choosing appropriate security control measures for various IoT systems. The three rightmost columns indicate the relation of each control measure with the security attributes. Though a single security level throughout the entire IoT system is advisable, a security designer may choose controls based on the individual (associated/ticked) attributes. For example: A system with High Availability, Low Integrity and Medium Confidentiality requirements would select High for Redundancy and Medium for Intrusion Detection and Prevention.

Note that the weighting and scoring of attributes is to be done by the user for the actual deployment and may vary depending on the organization's priorities and application scenario. It is out of scope for our proposed IoT security reference architecture.

Table 4 Mapping of Control Measures to Critical Activities (Super Table)

No.	Control Measures	Identifier	Critical Activity Identifier								Security attributes		
			SA	PP	CE	NT	DA	ST	CC	IO	C	I	A
1.	Entity Authentication	EA	✓		✓			✓	✓			✓	
2.	Key Management	KM			✓						✓	✓	
3.	Symmetric Key Encryption	SKE			✓			✓			✓		
4.	System Integrity / Hardware Security	SI		✓	✓		✓	✓				✓	
5.	Physical Security	PS	✓				✓	✓				✓	
6.	Access Control	AC	✓			✓	✓	✓			✓		
7.	User Authentication	UA						✓		✓		✓	
8.	Intrusion Detection and Prevention	IDP				✓					✓	✓	
9.	Redundancy	RDN	✓			✓		✓					✓
10.	Data Integrity / Message Authentication	DI			✓				✓	✓		✓	
11.	Data Sanity Checks	SAN		✓			✓			✓		✓	
12.	Configuration Management	CM							✓		✓	✓	✓
13.	Life Cycle Management	LCM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

4.5 Levels of Security Control Measures

Our reference architecture was designed to be applicable to all IoT domains. Therefore, it contains recommendations that use the currently highest feasible security measures (H), as well as the lowest security requirements that can be considered secure to date (L). Additionally, we have the intermediate level (M) with security measures ranging between the other two levels.

A system designer should be able to come to the conclusion that for critical infrastructure the level H might be required and chosen for a certain control measure and for a non-critical application environment such as a Smart Home the level M or L might be sufficient. Please refer to Use Case study in Chapter 6 and Annex 1 for user/implementor tips for further clarifications.

The following table classifies the security control measures into levels based on the severity of risks associated with each layer of the reference architecture. The table serves as a guideline for choosing the desired level of security while designing an IoT system [18, 19, 27-37].

Table 5 Levels Security Control Measures (Master Table)

Control Measures	Recommended Protection Mechanisms		EAL	IPDRR
Entity Authentication	**H**	1) Replay-resistant Challenge Response Authentication using Asymmetric keys **ACRYPT_H**[5], **HASH_L**[6] Or Symmetric keys **HASH_L**	4	Protect
	M	1) Replay-resistant Challenge Response Authentication using Asymmetric keys **ACRYPT_M, HASH_L** Or	3	

[5] Shown in Table 5
[6] Shown in Table 5

		Symmetric keys **HASH_L**		
	L	1) Replay-resistant Challenge Response Authentication using Asymmetric keys **ACRYPT_L, HASH_L** Or Symmetric keys **HASH_L**	_	
Key Management	**H**	Key Establishment: 1) Diffie Hellman **ACRYPT_H.** 2) a) PKI with public CA supporting certificate transparency b) PKI with internal CA Key Storage: 1) TPM / SE / TEE	4	Protect
	M	Key Establishment: 1. Diffie Hellman **ACRYPT_M** 2. a) PKI with public CA supporting certificate transparency b) PKI with internal CA Key Storage: 1) TPM / SE / TEE	3	
	L	Key Establishment: 1) Diffie Hellman **ACRYPT_L** 2) Pre-Shared Keys	_	

CHAPTER 4 | IOT SECURITY REFERENCE ARCHITECTURE

		2) Non-volatile Memory		
Symmetric Key Encryption	**H**	1) AES-256 Mode - GCM or CTR 2) ChaCha with Poly 1305(256)	4	Protect
	M	1) AES-192 Modes - GCM or CCM or CTR 2) ChaCha with Poly 1305 (256)	3	

	L	1) AES-128 Modes - GCM or CCM or CTR 2) ChaCha with Poly 1305(128), MUGI, SNOW 2.0		
	Lightweight Symmetric Key Cryptography for IoT is preferable once standardized [38].			
System Integrity / Hardware Security	**H**	1) Secure Boot and Remote Attestation using Trusted Execution Environment (TEE), Trusted Platform Module (TPM), or Secure Element (SE) 2) Hardware performance counters (HPCs)	4	Detect, Protect
	M		3	
	L	1) Secure Boot and Remote Attestation using Trusted Execution Environment (TEE), Trusted Platform Module (TPM), or Secure Element (SE)	_	
Physical Security	**H**	1) Tamper evident package 2) Tamper resistant package		Detect, Protect
	M			
	L	1) Simple tamper resistance setup (e.g. Tamper protection with locks)		
Access Control	**H**	1) Elimination of unnecessary ports, services, and protocols. 2) I. Mandatory Access Control II. Fine grained Mandatory Access Control 3) Attribute based Access Control 4) Network Segregation	4	Protect
		2) Discretionary Access Control 3) I. Role Based Access Control II. Identity Based Access Control		
	L	1) Elimination of unnecessary ports, services, and protocols.	_	

		2) Discretionary Access Control 3) I. Role Based Access Control II. Identity Based Access Control II. Group Based Access Control		
User Authentication	**H**	1) X.509 certificates with Digital Signatures **ACRYPT_H, HASH_H** 2) Authentication using passwords, personal identification numbers (PINs), tokens, biometrics, or in the case of multifactor authentication with level 4 assurance or some combination of the mentioned.	4	Protect
	M	1) X.509 certificates with Digital Signatures **ACRYPT_M, HASH_M** 2) Authentication using passwords, personal identification numbers (PINs), tokens or some combination of the mentioned.	3	
	L	1) Authentication using passwords, personal identification numbers (PINs), tokens.	_	
Data Integrity / Message Authentication	**H**	1) Digital Signatures **ACRYPT_H** 2) HMAC **HASH_L,** 256 bits key length.	4	Detect
	M	1) Digital Signatures **ACRYPT_M** 2) HMAC **HASH_L,** 192 bits key length	3	
		2) HMAC **HASH_L,** 128 bits key length		
Redundancy	**H**	1) Duplication (Triplication) of critical components, automatic failover and error reporting.	_	Detect, Response, Recovery
	M	1) Duplication of critical components, automatic failover and error reporting.	_	

	L	1) Significant local storage with redundant data storage.	_	
Data Sanity Checks	**H**	1) Input Validation 2) Behavioural Monitoring of incoming data	4	Detect, Response
	M	3) Time Synchronization: Network Time Protocol (Best Practices: Draft- ietf-ntp-bcp-08)[7] Precision Time Protocol (IEEE 1588-2008)[8]	3	
	L	1) Input Validation	_	
Intrusion Detection / Prevention	**H**	1) Host based intrusion detection 2) Network based intrusion detection. a. Honeypots and Honeynets with Threat intelligence feedback for malware analysis. b. Anomaly based and signature-based intrusion detection. c. Deep Packet Analysis 3) Network Firewall 4) End device monitoring[9]	4	Detect, Protect, Response, Recovery
	M	1) Network based intrusion detection. 2) Anomaly based and signature based 3) Reputation based lists 4) Network Firewall 5) End device monitoring[10]	3	
	L	1) Network based intrusion detection. a. Anomaly based and signature-based intrusion detection. b. Reputation based lists	_	

[7] Draft still being updated by working groups.
[8] Draft still being updated by working groups.
[9] Refer Helper Table (Table 6)
[10] Refer Helper Table (Table 6 and 7)

		2) Network Firewall		
Configuration Management	**H** **M** **L**	1) Centralization of policy and tailoring configurations according to device (system) role. 2) Patch management. 3) Backup Management 4) User Account/Identifier Management (e.g. credentials, privileges etc.) 5) System Settings Management 6) Network Management (Best practices)		Protect, Response, Recovery
Life Cycle Security	**H** **M** **L**	1) Design and develop the system using a secure system engineering approach 2) Implement and maintain the system with components from a secure supply chain, with no known vulnerabilities 3) Vulnerability and Penetration Testing and proper vulnerability disclosure 4) Security Audits for the entire system as applicable 5) Sanitization of devices/systems of security data and sensitive user data, before reuse or disposal		Identify, Detect, Protect, Response, Recovery

CHAPTER 4 | IOT SECURITY REFERENCE ARCHITECTURE

Table 6 Helper Table for Hash Functions and Asymmetric Key Cryptography

HASH – cryptographic hash functions	
H	SHA3-512
	SHA-512

M	SHA3-384
	SHA-384
L	SHA3-256
	SHA-256

ACRYPT – asymmetric cryptography	
H	Elliptic curve P-521
	RSA (15360 bit key length)
M	Elliptic curve P-384 or Curve448
	RSA (7680 bit key length)
L	Elliptic curve P-256 or Curve25519
	RSA (3072 bit key length)

Table 7 Helper Table Monitoring

End Device Monitoring
1) Profiling each node for identification, capacity, location and performance summary. 2) Tamper-proof logs of the system status. 3) Maintaining a record for all nodes connected to or disconnected from a certain node with routine updates. 4) Log of authentication attempts. 5) Revoking the connectivity of an infected device within pre-determined safe time limit (precondition). 6) Patch status monitoring. 7) Application Errors Tracking and Reporting.

In this chapter, we show IoT uses cases where the entire exercise of identifying recommendations based on the security reference architecture has been performed. The protocols selected are based on the IoT system design and may vary based on the discretion of the system designer even while maintaining the same level of security.

For each of the two case studies, we invented a limited system architecture which includes system specifications (possible functionality, devices connected, network connectivity[11], logical data flow, data processing and service framework). Based on the system specifications we performed risk analysis and identified the security requirements. The recommendations are finally based on our idea of the security requirements of the IoT system. This chapter is meant to be an exercise to demonstrate the step-by-step recommendation generation procedure and usability of the reference architecture. The user/implementor must develop the system architecture based their own system specifications.

5.1 Use Case Study 1 – Smart Lamp Posts

The first IoT system to be studied in this section is the Smart Lamp Post. Multipurpose Smart Lamp Posts are a global trend nowadays and are already on the way of implementation in the cities of South Korea and China.

South Korea

While the earlier versions of Smart Lamp Posts just served the purpose of controlled illumination, Lamp posts fitted with an array of smart IoT sensors serve purposes like weather monitoring, traffic/crowd monitoring, surveillance etc. in cities of South Korea[12]. The government plan is to integrate ICT with street lighting to offer road environment information to surrounding pedestrians or vehicles with Infrastructure to Thing (I2X) technologies to reduce traffic accidents. The main goal is to avoid frequent accidents in areas such as crosswalks, intersections and tunnels by sending digital signals from the smart lamp post fixtures to vehicles or pedestrians so that they can react immediately to emergency road situations. The Ministry of Land, Infrastructure and Transport is currently overseeing the

[11] This is a network protocol agnostic work. The network protocols shown in the diagrams serve as example and in no way suggest the use of those specific protocols.

[12] http://www.molit.go.kr

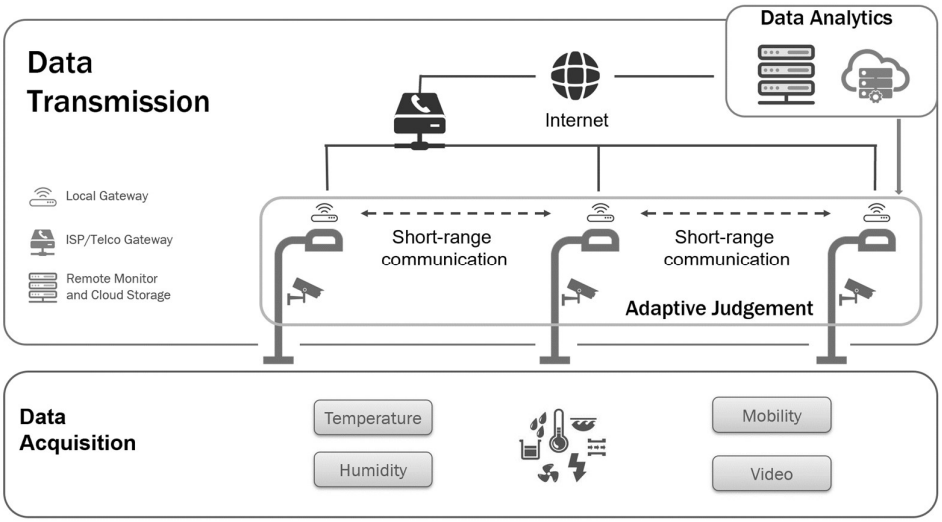
multi-ministry project and outlining the strategy of providing a street lighting platform for the multi-sensor convergence street lighting system within 2023.

China

Intelligent streetlights have been deployed experimentally in various cities across China, for example Chongqing and Yangzhou[13]. Apart from energy saving the multipurpose sensor fitted lamp posts can monitor *environmental data* like temperature, humidity, pressure, wind speed, wind direction, *ambient data* like noise and pollutant level, *monitor* on-road vehicles and pedestrians. The data collected from different smart lamp posts all over the city, can be transferred to a secure cloud-based platform for analysis and related applications. The cloud platform is also used for easy remote management of the smart lamp posts. API access to the platform is also planned to be provided to the end users of some Telcos (e.g. China Mobile), so that they can integrate the data into their own platforms and have unique services.

With the enhanced capabilities of the Smart Lamp Posts it requires a comprehensive idea of the entire IoT ecosystem around them and a detailed understanding of their security needs. This section will focus on the understanding of the entire Smart Lamp Post operating environment. A Smart Lamp Post environment, catering to the different stages of the system pipeline is shown in figure 3 (the conceptual model is created by studying a few real-world deployments as provided in the examples above).

Figure 3 Conceptual Model of the Smart Lamp Post IoT Systems



[13] https://www.gsma.com/iot/wp-content/uploads/2018/03/iot_china_mobile_lighting_04_18.pdf

Things-Centric View

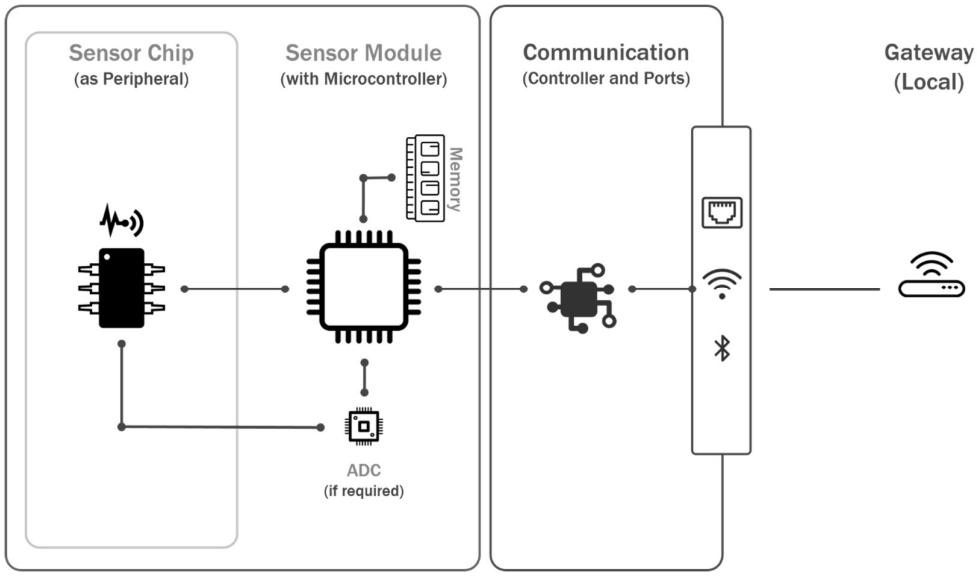
The data acquisition stage involves different sensors. In our study we consider that a Smart Lamp Post IoT system may require the following categories sensors to perceive and monitor the surroundings [39],[40].

A. **Sensors by Functionality** – According to the functionality, the sensors in a Smart Lamp Post can be categorized as Environment, Ambient and Activity Sensors.

B. **Sensors by Design** – According to design, the sensors may be reclassified according to form factor as
   - Slave Devices – Sensor chips with no microcontroller or communication.
   - Sensor Modules – Individual Sensor IC along with on-board microcontroller, generally with some communication capability.
   - Integrated Environments – Multiple Sensors in the form of a Single IC, or a plug-and-play Module with Multiple Sensors, along with an on-board microcontroller and communication capability.

After scoping the types of sensors, the understanding of the different architectural views is important to form an idea about the granularity of the devices and overall data acquisition. A possible connectivity model of a sensor chip to a microcontroller/microprocessor board is shown in the figure 4. If the sensor is in the form of a module/plug and play device, there is no need to connect them to other microcontroller boards as they can perform communication independently.

Figure 4 Thing-Centric Architectural View (Sensor Chip with Microcontroller)

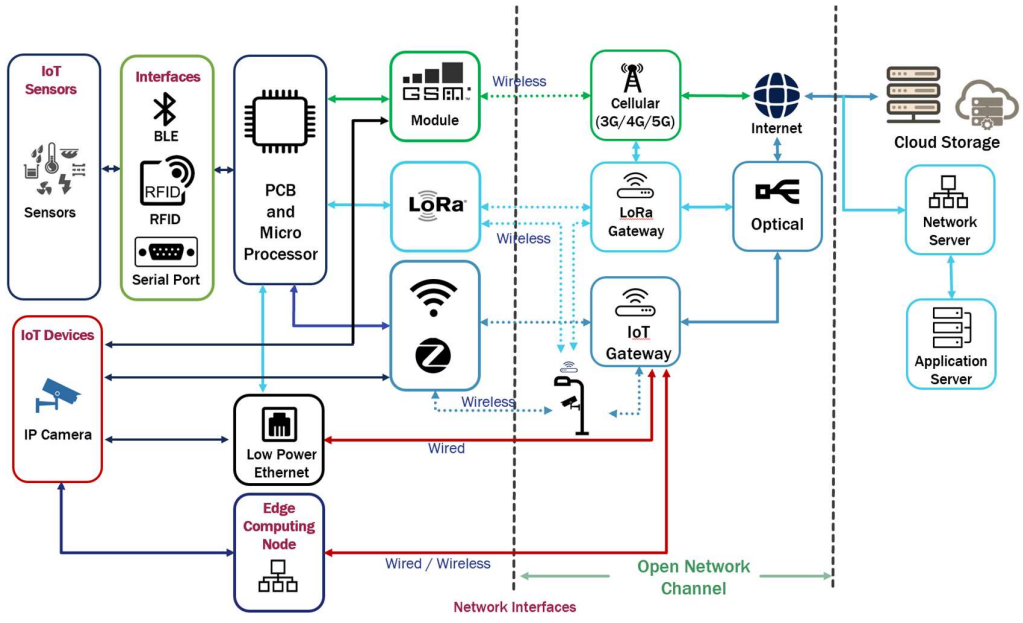


From the Things-centric view we understand that the end devices in a smart lamp posts environment range from low end embedded devices (sensors integrated with inexpensive microprocessors) with no networking capability of their own to a decent computation and communication capable devices like the IP cameras. From the Things-centric view we understand that heterogeneity, resource and communication constraints must be kept in mind for the selection of the critical activities and the corresponding security controls.

Network-centric view

Continuing with our bottom up approach comprehending the Network-centric architectural view is important to understand the security issues associated with data transmission [39]. The figure below presents the possible components, communication technologies and data flow patterns shaped into the Network-centric architectural view. From the Network-centric view we understand that limited range, low power proximity networks generally constitute all local connections to serve the specialized needs of this IoT system. Wide area networks that connect the proximity networks to the internet, are shared general-purpose networks provided by Telcos. Proximity networks are managed by the IoT application while the wide area networks may not be. The Network-centric view also provides us with a detailed understanding of the risks and the attack surfaces/targets as discussed below.

Figure 5 Network-Centric Architectural View



Risk Analysis

To comprehend the security issues and respective countermeasures a detailed idea about the attack surfaces of the Smart Lamp Posts is needed. Table 1 summarizes the attack categories, the components/parts of the Smart Lamp Post IoT infrastructure they effect and compromised security principles [41],[42]. The risk analysis is easier if the Network-centric view is broken down to proximity networks and WAN. The same approach is followed in this use case study to have a detailed risk analysis. We are using STRIDE model here, but other methodologies of threat modelling exist, and the security designer may use any of the standard models if it suits the specific use case.

Table 8 Risk Analysis

Threat	Attack type	Target	Compromised CIA principle
Spoofing	• Sensor identity spoofing • Provide wrong sensor input	• Between Devices and internet • Between gateway and network interface • Between cloud/server and device	Integrity, Availability
Tampering	• Physical Tampering • Uncontrolled resource consumption • Manipulation of transmitted packets • Malicious firmware update • Stored data manipulation	• Devices • Communication channels • Cloud Storage	Integrity, Confidentiality, Availability
Repudiation	No significant repudiation attacks.		
Information Disclosure	• Passive (listening to the message transmission) • Active (exploiting bugs, reading error logs etc.)	• Device to internet • Internet to cloud storage	Confidentiality

Denial-of-Service	• Jamming • Flooding • Manipulation of routing paths • Packet drop with malicious intent	• Network interface to IoT gateway • Internet to IoT gateway	Availability
Elevation of privilege	• Malware elevating access rights/privileges • Normal users can access resources/ contents reserved for administrative users	• Devices • Cloud storage	Confidentiality, Integrity, Availability

Activity-centric view

Due to complexity and capability constraints of a practical system environment the threats and security requirements may vary. In the following section we present a sample security recommendation based on our understanding of the system specifications, security requirements and impacts of attacks (refer FIPS 199) on Smart Lamp Posts IoT application

Recommendation Generation for Smart Lamp Posts

System Considerations (Functional):

A. To collect environmental data using various low-end sensors for weather report and forecasting
B. To record video of traffic to reconstruct accidents

System specifications (Example):

1) Temperature and humidity sensors integrated with a microcontroller mounted on the lamp post
2) IP camera mounted on the lamp post using TLS 1.3 secured channels to connect to the backend
3) Wired ethernet connectivity with the microcontroller and the IP camera (each lamp post has a wired connection to one of the local gateways)
4) Local gateways are connected to the public internet providing an IPSEC VPN channel to the backend
5) The collected data is stored and made available via web in a cloud platform.

Basic Privacy Considerations:

1) Environment sensing data does not contain any Personally Identifiable Information (PII).
2) Video surveillance data will contain personally identifiable information e.g. faces, license plates, travel patterns, specific buildings etc.

Basic Privacy Recommendations:

1) To minimize unnecessary collection of PII, each camera is restricted to the relevant field of view.
2) Additionally, each camera performs preprocessing, detecting PII (e.g. faces and license plates) and rendering them unrecognizable.

Security Considerations:

1) Weather data is publicly available and therefore ***Confidentiality*** is not required.
2) ***Integrity*** and ***Availability*** of weather data is beneficial but not critical.
3) As the preprocessed video may still contain PII ***Confidentiality*** is of high importance.
4) ***Integrity*** of the video stream is of critical importance as it may be used as evidence during legal proceedings.
5) As the video is not used for real time decision-making, the ***Availability requirements are low. Rather than quick access, retention of data is important.***

Table 9 Security Needs for Assets (Based on Things and Network-centric Knowledge)

Assets	C	I	A	Rationale
Microcontroller	L	L	L	We consider that sensors integrated with the microcontroller do not collect sensitive data.
IP camera	M	M	L	Camera acquires sensitive information in the form of audio and video. Leaking of the data may lead to access of PII. A privacy breach may result in serious consequences for a person and significant fines for the operator. Tampering of data may lead to false information transmission that may be used as evidence in courts risking serious impacts on the defendant. So, confidentiality and integrity need to be medium, but availability

				may be Low as failing nodes can be easily detected and the probability of an accident occurring during the downtime is low.
IoT Gateway	L	L	L	IoT Gateway is an important asset for smooth service of the entire system. It may transmit individual or aggregated sensor data that has low confidentiality, integrity and availability requirements. The fact it also transmits the high confidentiality and integrity video data does not increase the gateway's security requirements, as these data are already protected by a TLS channel between the camera and the backend.
Cloud	M	M	M	Sensitive data storage and processing but no critical decision making. Cloud infrastructure may also provide platform for interoperability between different agencies. So, we consider all the three security attributes to be medium.

Organizations should conduct a more detailed analysis of security needs based on a weighted matrix of impact and likelihood satisfying their goals. We here follow a simple walkthrough to demonstrate the usability of the architectural components.

Security Recommendations:

Things-centric and Network-centric views provided a broader perspective to understand security requirements. However, to customize the security recommendations we need to refer the Activity-centric view. Note that the table below lists the security recommendations based on our system specifications based on the activities that a smart lamp post system may perform. The specifications may vary from organization to organization and from application to application. If the specifications vary the risk assessment may also vary. While using the reference architecture caution must be adapted to choose controls based on the impacts of security incidents on a specific system. The needs of one organization may vary vastly from another and thus the security professionals of one organization may consider risk tolerance level higher than the other. The proposed security reference architecture is generic enough to accommodate the High, Medium and Low security requirements of generic IoT systems, but it is entirely up to the organization to decide how to manage, mitigate or distribute the risks.

Sl. No	Devices/Infrastructure {Critical Activity ID Obtained from Activity-centric view of the application}	Security Requirement {ID-Level Obtained from our analysis of Security needs}	Control Measures	Rationale
1.	**Microcontroller {SA}**	EA - Low	Network Address (MAC)	Low CIA requirement. Wired connection (Internal/Underground)
		PS - Low	Simple tamper resistance setup inside the lamp post.	To mitigate inconvenience caused due to physical damage.
		AC- Low	Elimination of unnecessary ports.	System hardening.
		RDN - Low	NIL	No specific control measures as data is already redundant at adjacent lampposts.
2.	**IP Camera {SA, PP, CE}**	EA - Medium	TLS 1.3 (X.509 certificates EdDSA using SHAKE256 and Curve448 (Ed448), private CA)	TLS 1.3 provides digital signature based mutual authentication, replay-resistance, forward secrecy.
		KM - Medium	Key Establishment: TLS 1.3 ECDH on Curve448 (X448)	Connections restricted to the IoT ecosystem.

			Setup of private CA. Key Storage – TPM	
		SKE - Medium	TLS 1.3 (AES-192 - GCM)	High confidentiality requirement.
		DI - Medium	TLS 1.3 (AES-192 - GCM)	High integrity requirement. AES-192-GCM provides authenticated encryption.
		PS - Medium	Tamper evident packaging.	To prevent unauthorized physical access as the camera is a field device and handles highly sensitive data.
		AC - Medium	Discretionary Access Control Elimination of unnecessary ports, services, protocols.	Limit access to highly sensitive data.
		RDN - Low	Buffer.	Failing nodes can be easily detected. Probability of an accident occurring during the downtime is low and required video material of proximate cameras may be sufficient.
		SI - Medium	TPM, remote attestation.	Hardware root of trust for system integrity.

		IDP - Medium	Log of authentication attempts. Tamper-proof logs of the system status (TPM, remote attestation.).	To detect/protect endpoint from compromise.
		SAN - Medium	Behavioural monitoring (Video liveness detection based on motion events). Time Synchronization.	Optical input validation should take place before preprocessing, as masking might reduce important cues.
3.	**Local Gateway {CE, NT} (Provides VPN tunnel between the device and the backend)**	EA - Low	IKEv2 X.509 certificate authentication using ECDSA on Curve25519 and SHA-256 (private CA)	Sensor data only has low confidentiality, integrity and availability requirements. Endpoints for video are mutually authenticated by TLS 1.3.
		KM - Low	Key Establishment: IKEv2 ECDH on Curve25519 (X25519) Key Storage – TPM.	Sensor data only has low confidentiality, integrity and availability requirements.

		SKE - Low	IPSec (AES-128-GCM)	Sensor data only has low confidentiality requirement. Forwarded video stream integrity protected by TLS 1.3.
		DI - Low	IPSec (AES-128-GCM)	AES-128-GCM provides authenticated encryption. Sensor data only has low integrity requirement. Forwarded video stream integrity protected by TLS 1.3.
		AC - Low	Role based access control. Elimination of unnecessary ports, services, protocols. Network segregation.	Limit access to aggregated data.
		RDN - Low	NIL	Failing nodes can be easily detected. Probability of an accident occurring during the downtime is low.
		IDP - Low	Anomaly based traffic analysis to create alarms Network Firewall	In normal operations traffic patterns are expected to be same which eases the detection of anomalies. Access to the VPN gateway should be limited

				to the connected lamp posts.
		SI - Low	TPM secure boot.	Hardware root of trust.
4.	**Cloud (Linux) {CE, ST, CC, IO} (Collects, stores data and provides it via a web server)**	EA – High / Low for the VPN	TLS 1.3 (X.509 certificates EdDSA using SHAKE256 and Curve448 (Ed448), private CA) IKEv2 X.509 certificate authentication using ECDSA on Curve25519 and SHA-256 (private CA)	High for video stream from camera. Low for sensor data.
		KM – Medium / Low for the VPN	Key Establishment: TLS 1.3 ECDH on Curve448 (X448) / IKEv2 ECDH on Curve25519 (X25519) Private CA for internal communication and Public CA for external connection.	Private CA for internal communication and Public CA for interoperability between agencies.

			Key Storage – TPM, Virtual TPM.	
		SKE - Medium / Low for the VPN	TLS 1.3 (AES-192 – GCM) IPSec (AES-128-GCM) Storage: dm-crypt with AES-256-CBC	Storage of raw data, credentials, processed metadata etc. Highly sensitive data requires storage encryption to ensure confidentiality if an adversary gains physical access to disks.
		DI – Medium/ Low for the VPN	TLS 1.3 (AES-192 – GCM) IPSec (AES-128-GCM)	AES GCM provides authenticated encryption. High for video data. Low for sensor data.
		PS - Medium	Tamper evident design.	To prevent unauthorized physical access as the infrastructure has high integrity requirements.
		SAN - Medium	Input Validation Behavioural monitoring and alarm generation Time Synchronization.	Sensor input can be more easily detected as all sensor data are present. As correlations between adjacent sensors is to be expected, malfunctioning or spoofed sensors can be detected.
		AC - Medium	Role based access control. Elimination of unnecessary ports, services, protocols.	Limit access to highly sensitive data.

		RDN - Medium	RAID 6 storage. Multiple internet connections. Automatic Backup.	The backend should stay permanently operational even if components fail.
		SI - Medium	TPM, Virtual TPM.	Hardware root of trust for system integrity of host and virtual machines.
		IDP - Medium	Log of authentication attempts. Tamper-proof logs of the system status (TPM). Anomaly and signature based detection. Maintaining a record for all nodes connected to or disconnected from with routine updates.	Attacks on the backend infrastructure require timely detection.
		UA - Medium	TLS 1.3 (Client - authenticated, X.509 certificates EdDSA using SHAKE256 and Curve448 (Ed448), public CA)	Agencies accessing the sensitive data can be assumed capable of handling client certificates. End users accessing public weather data do not need to be authenticated.

5.2 Use Case Study 2 – Smart Metering as an Enabler of Smart Home

In this section we study an IoT system which is a combination of two IoT systems: Smart Metering and Smart Home. Smart meters have been adopted in different countries aiming to help in efficient energy consumption and billing. We discuss the deployment case studies from UK and Australia and present a generic example application where the smart meter includes the control functionality of a smart home gateway.

United Kingdom[14][15]

The UK government backed organization *Smart Energy GB* has a plan to install smart meters in every home by 2020 as a part of implementation program (SMIP). The installation aims for the following capabilities:

1. *Smart prepayment:* Develop the capability to deliver smart prepayment requirements, starting from the top-up process through multiple payment channels and, eventually, the top-up landing on the smart meters.

2. *Debt management:* Develop capability to manage the debt lifecycle by enhancing smart metering functionalities for eligible customers with debt. This will involve auto-reconciliation of prepayment top-ups and instalments towards debt recovery for the utility.

3. *Home automation:* Develop home automation/Internet of Things (IoT) capabilities to create a comprehensive customer experience integrating control and administration of customer's smart meters, smart domestic appliances, smart electrical installations etc.

4. *Alarms and alerts:* Mechanisms to send real-time (high priority) and daily summary (low priority) alarms and alerts to utilities. This allows utilities to proactively predict and resolve any issues with smart metering assets remotely using over-the-air transfer capabilities.

Whilst the initial prototypes of rolled out smart meters used mobile phone networks, the current models use a tailored, secure network structure.

---

14 https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/426135/Smart_Metering_System_leaflet.pdf

15 https://www.cognizant.com/whitepapers/scaling-up-smart-meter-operations-challenges-and-the-way-forward-for-uk-energy-utilities-codex2184.pdf

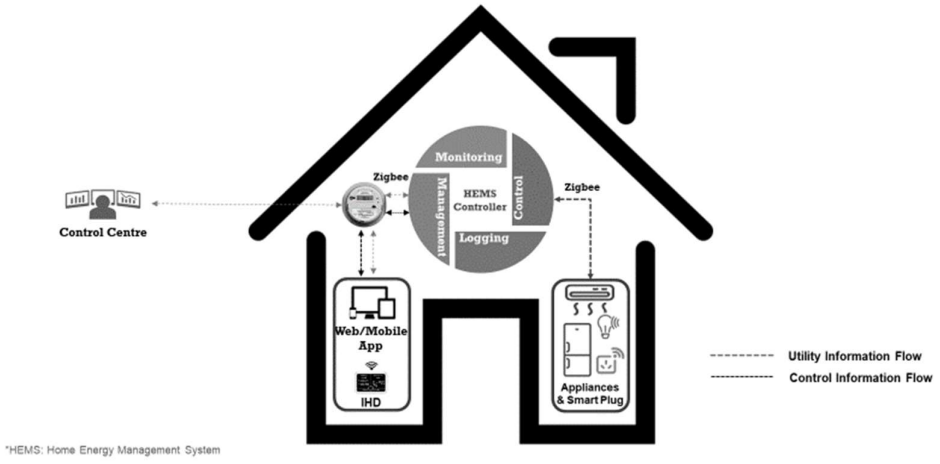
Australia

Smart electricity meters are installed in 70% of Australian homes and businesses as of 2019[16]. Australian Communications & Media Authority (ACMA) claim that the smart meters in Australia provide the following benefits:

1. *Improves performance, including reliability and quality of supply, and permit fault identification and network load management.*
2. *Enables a safe, competitive, open and fair market for demand side services*
3. *Enables additional functions such as remote energisation and de-energisation and appliance control.*
4. *Can link to household devices such as through a Home Area Network (HAN) and In-Home Display (IHD) to enable instant access for the consumer to their electricity use profile.*

The Australian energy regulators also claim have privacy preserved data communication[17] from individual meters to the retailer. A conceptual model showing the integration of the operations of Smart Home and the Smart Metering systems is presented below [43],[44],[45],[46].

Figure 6 Conceptual Model for Smart Metering As an Enabler of Smart Home



[16] https://www.energynetworks.com.au/smart-metering
[17] https://www.sa.gov.au/topics/energy-and-environment/meters-and-bills/smart-meters

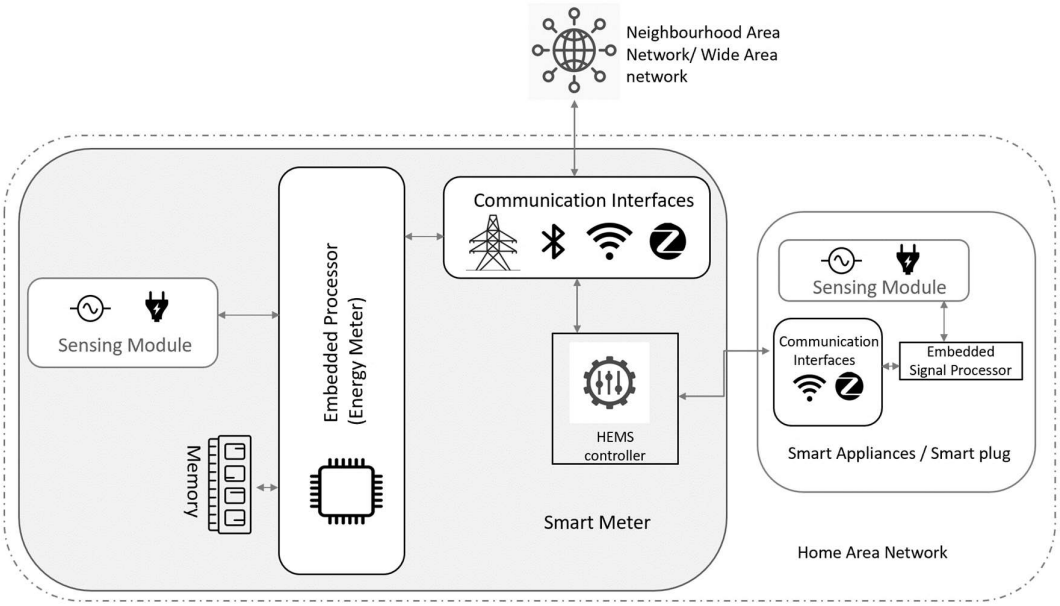
IoT Devices and Thing-Centric View

Smart metering together with smart home systems provide advantages to utilities and customers alike [43]. Considering the Advanced Metering Infrastructure (AMI) along with the connected home IoT structure we identify the following to be the possible IoT devices:

I. Smart Meters: Intelligent meters capable of communication. A smart meter consists of sensors to track the current flow, a microcontroller to control metrology or for home appliance management functions, and communication module.
II. Smart home appliances: Intelligent home appliances capable receiving control signals from smart meters, processing and acting according to the signal.
III. Smart Plug: Smart plug can send control signals to dumb appliances that plugs into a standard wall socket. Smart meters will be able to manage the devices through smart plug.

The Things-centric architectural view shows that the sensing module is connected to the embedded processor which in turn is connected to a communication interface. The figure below depicts the Things-centric architectural view. In this study we consider that smart meter may be able to send control signals to the appliances, so we consider that Home Energy Management System (HEMS) controller is a part of the Smart Metering device.

Figure 7 Thing-centric Architectural View

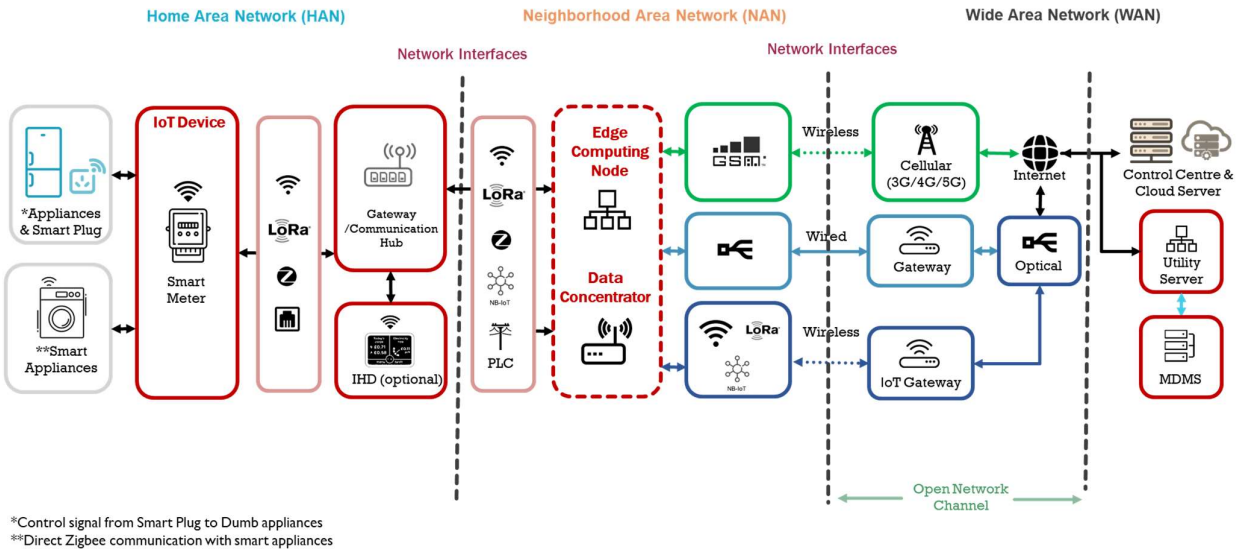


From the Things-centric view we understand that in this IoT environment integrity is of prime importance. Also, in this IoT system, resources, communication, security capabilities of the end devices may vary to a great extent from one home environment to another.

Network-centric View

The figure below presents the possible components, communication technologies and data flow patterns shaped into the Network-centric architectural view. Similar to the previous use case there are proximity networks (HAN, NAN) and Wide Area Networks. From the Network-centric view we understand that unlike smart lamp posts in this application the end devices are placed within a home perimeter thus providing better physical security. But the edge devices and the data concentrators playing an important role in this application are deployed in the fields giving threat actors access to accumulated and pre-processed data.

Figure 8 Network-centric Architectural View



Risk Analysis

The following table summarizes the attacks and the components/parts of the Smart Metring (as an enabler of Smart Home) [47],[48] system they affect based on the Things-centric and Network-centric views.

Table 10 Risk Analysis

Threat	Attack Type	Target	Compromised CIA principle
Spoofing	• Meter identity spoofing • Provide misleading information for routing • Incorrectly attribute billing	• Between meter(s) and internet • Between gateway and network interface • Between cloud/server and device	Integrity, Availability
Tampering	• Manipulation of resources (e.g. Meter memory, Battery power etc.) • Manipulation of packets • Malicious Firmware update • Physical damage • Storage Tampering	• Smart Meters and smart appliances (Hardware) • Data concentrators • Between Smart Meter and internet • Between Smart Meter and remote centre /cloud	Confidentiality, Integrity, Availability
Repudiation	• Manipulation to incorrectly credit the actual usage	• Smart Meter and appliances	Integrity
Information Disclosure	• Interception of and modelling of PII with usage • Leakage of billing information • Leakage of stored data / credentials	• Smart Meter, • Data concentrator, • Communication channels, • Cloud	Confidentiality

Denial-of-Service	• Jamming • Flooding • Grid Disruption (Simultaneous on-off of a significant number of meters) • Manipulation of routing paths • Packet drop with malicious intent	• Internet to home gateway • Between meter, concentrators and internet. • Between meter and appliances.	Availability
Elevation of Privilege	• Malware elevating access rights/privileges • Unauthorized users can access resources/ contents reserved for administrative users of Meter Data Management System (MDMS)	• Smart Meter • Control centre • Cloud storage	Confidentiality, Integrity, Availability

The key security objectives are found to be:

A. *Mutual authentication*,
B. *Tamper evidence,*
C. *Ensuring availability (information as well as control)*,
D. *Access Control,*
E. Encryption of usage information and Key Management.

Activity-centric view

Due to complexity and interoperable nature of the practical system environment the threats and security requirements may vary. In the following section we present a sample security recommendation based on our understanding of the system specifications, security requirements and impacts of attacks (refer FIPS 199) on Smart Metering application.

Recommendation Generation for Smart Metering as an Enabler of Smart Home

System Considerations (Functional):

- Measure power consumption and calculate billing data.
- Communicate with appliances for efficiency of consumption.
- Communicate with a web application (via in home display, smartphone etc.) to show the consumption information to the customer.

System specifications (Example):

- Smart meter installed in a building premise.
- Smart meter and appliances connectivity via wireless communication.
- Wired PLC connectivity between meter and local gateway.
- Local gateway and data concentrators connected via wired ethernet.
- Data concentrators have edge computing capabilities to perform preprocessing.
- Data concentrators are connected to the public internet using standard protocols.
- The collected data is stored and made available via web in a cloud platform.
- All communication channels are secured by TLS 1.3.

Basic Privacy Considerations:

Power consumption is related to the general behaviour of a consumer and may thus contain PII (e.g. home occupancy time).

Basic Privacy Recommendations:

- To reduce the impact of PII data is down sampled on the smart meter (down sampling is a form of preprocessing).
- Data anonymization approach utilizing pseudonyms rather than the real identities of consumers.

Security Considerations:

- For the billing data Confidentiality, Integrity and Availability are important.
- Controlling the smart appliances primarily requires Integrity and Availability.

Table 11 Security Needs of Assets (Based on Things and Network Centric Knowledge

Assets	C	I	A	Rationale
Smart Meter	M	M	L	Confidentiality is Medium as the meter collects sensitive customer data. Leaking of these data may result in consumer safety compromise if an adversary can model consumption behaviour. Integrity is Medium as manipulation of meter data may lead to financial loss or personal safety compromise. An adversary may also use a meter as an entry point to launch further attack on smart appliances compromising consumer security. Assuming that an unavailable smart meter will not cut power, availability is Low as lack of availability will hinder data to/from a single household and can be easily detected within a short span.
Local Gateway	M	M	L	Confidentiality is Medium as the local gateway transmits sensitive consumer data. Integrity is Medium as the manipulation of gateway data may lead to financial loss. Availability is Low as lack of availability will hinder data to/from a single household and can be easily detected within a short span.
Data Concentrator	M	M	M	All three attributes are considered Medium as the data concentrator collects and transmits data from multiple households in a neighbourhood.
Cloud/Utility Control Centre	M	M	M	Storage and processing of personal information at Cloud is the reason for considering Confidentiality and Integrity to be Medium. Availability is considered Medium as unavailable billing data for a certain time will not have much impact on consumer.

Security Recommendations:

As shown in the previous section Things-centric and Network-centric views may not be enough to understand security requirements or recommend customized security control measures. The Activity-centric view is thus referred for associating the critical activities to the necessary security control measures.

Sl. No.	Devices/Infrastructure (Critical Activity ID)	Security Requirement	Control Measures	Rationale
1.	**Smart Meter {SA, PP, CE, IO}**	EA - Medium	TLS 1.3 (Client - authenticated, Elliptic Curve P-384)	TLS 1.3 provides digital signature based mutual authentication, replay-resistance, forward secrecy.
		KM - Medium	TLS 1.3 (ECDHE P-384). Public Root CA. Key Storage – TPM	Interoperability required between different stakeholders.
		SKE - Medium	TLS 1.3 (AES-192 - GCM)	Medium confidentiality requirement.
		DI - Medium	TLS 1.3 (AES-192 - GCM)	Medium integrity requirement. AES-192-GCM provides authenticated encryption.
		PS - Medium	Simple tamper resistant setup.	Medium integrity requirement but the device

				is present within a home boundary.
		AC - Medium	Attribute based Access Control Elimination of unnecessary ports, services, protocols.	Limit access to highly sensitive data.
		RDN - Low	Buffer and local storage.	Only for network failures for a short period of time. In case of total failure of the device it can be easily detected by the backend and the revenue loss is not significant.
		SI - Medium	TPM.	Hardware root of trust for system integrity.
		IDP - Medium	Log of authentication attempts. Tamper-proof logs of the system status (TPM).	To detect/protect endpoint from compromise.
		SAN - Medium	Input Validation Behavioural monitoring and alarm generation (frequencies above half the down sampled sample rate). Time Synchronization.	Input data is expected to show distinctive patterns so that malfunctioning appliances can be detected.

2.	**Local Gateway {NT, CC, CE}**	AC - Medium	Identity based Access Control Elimination of unnecessary ports, services, protocols.	Confidential data is otherwise protected.
		RDN - Low	NIL	In case of total failure of the device it can be easily detected by the backend and the revenue loss is not significant.
		IDP - Low	Anomaly based detection (Traffic Analysis) Network Firewall	In normal operations traffic patterns are expected to be same. Strong protection mechanisms already considered for endpoint.
		EA - Medium	TLS 1.3 (Client - authenticated, Elliptic Curve P-384)	TLS 1.3 provides digital signature based mutual authentication, replay-resistance, forward secrecy.
		KM - Medium	TLS 1.3 (ECDHE P-384). Public Root CA. Key Storage – TPM	Interoperability required between different stakeholders.
		SKE - Medium	TLS 1.3 (AES-192 - GCM)	High confidentiality requirement.

		DI - Medium	TLS 1.3 (AES-192 - GCM)	High integrity requirement. AES-192- GCM provides authenticated encryption.
		SI - Medium	TPM.	Hardware root of trust for system integrity.
3.	**Data Concentrator {CE, NT, PP}**	EA - Medium	TLS 1.3 (Client - authenticated, Elliptic Curve P-384)	TLS 1.3 provides digital signature based mutual authentication, replay-resistance, forward secrecy.
		KM - Medium	TLS 1.3 (ECDHE P-384). Public Root CA. Key Storage – TPM	Interoperability required between different stakeholders.
		SKE - Medium	TLS 1.3 (AES-192 - GCM)	High confidentiality requirement.
		DI - Medium	TLS 1.3 (AES-192 - GCM)	High integrity requirement. AES-192- GCM provides authenticated encryption.
		AC - Medium	Role based Access Control Elimination of unnecessary ports, services, protocols.	Limit access to highly sensitive data.
		RDN - Medium	Connection of home gateways to multiple aggregators for failover.	Aggregators should not be single point of failures.

		IDP - Medium	Signature based detection Network Firewall	Interesting target for a variety of threat actors. Signature-based detection will enable identification of malware samples.
		SAN - Medium	Input Validation Behavioural monitoring and alarm generation. Time Synchronization.	Input data is expected to show distinctive patterns so that malfunctioning smart meters can be detected.
4.	**Cloud (Linux) {CE, ST, CC, DA, IO} (Collects, stores, analyzes data and provides it via a web server)**	EA – Medium	TLS 1.3 (X.509 certificates ECDSA using SHA-384 Curve P-384, public CA)	TLS 1.3 provides digital signature based mutual authentication, replay-resistance, forward secrecy.
		KM – Medium	TLS 1.3 (ECDHE P-384, Public CA) Key Storage – TPM, Virtual TPM.	Interoperability required between different stakeholders.
		SKE – Medium	TLS 1.3 (AES - 192 - GCM) Storage: dm-crypt with AES-256-CBC	Storage of raw data, credentials, processed metadata etc. Highly sensitive data requires storage encryption to ensure confidentiality if an adversary gains physical access to disks.
		DI – Medium	TLS 1.3 (AES-192 - GCM)	AES-GCM provides authenticated encryption.

		PS - Medium	Tamper evident design.	To prevent unauthorized physical access as the infrastructure has high integrity requirements.
		AC - Medium	Role based Access Control Elimination of unnecessary ports, services, protocols.	Limit access to highly sensitive data.
		RDN - Medium	RAID 6 storage. Multiple internet connections. Automatic Backup.	The backend should stay permanently operational even if components fail.
		SI - Medium	TPM, Virtual TPM.	Hardware root of trust for system integrity of host and virtual machines.
		IDP - Medium	Log of authentication attempts. Tamper-proof logs of the system status (TPM). Anomaly and signature-based detection. Maintaining a record for all nodes connected to or disconnected from with routine updates.	Attacks on the backend infrastructure require timely detection.

		SAN - Medium	Input Validation Behavioural monitoring and alarm generation Time Synchronization.	Input data is expected to show distinctive patterns so that smart meters malfunctioning or spoofing can be detected.
		UA - Medium	TLS 1.3 (X.509 certificates ECDSA Curve P-384) Client authentication on web application layer.	End users accessing public billing data are not used to handle client certificates authenticated.

For verification, additionally to the two case studies presented in Chapter 6, the proposed IoT security reference architecture was applied to a Smart Home use case study provided by IMDA Singapore. The case study includes a system architecture (Things + Network-centric) and detailed risk assessment presented below, allowing to directly apply the security recommendations based on the Activity-centric view.

Figure 9 System Architectural of Home Control System

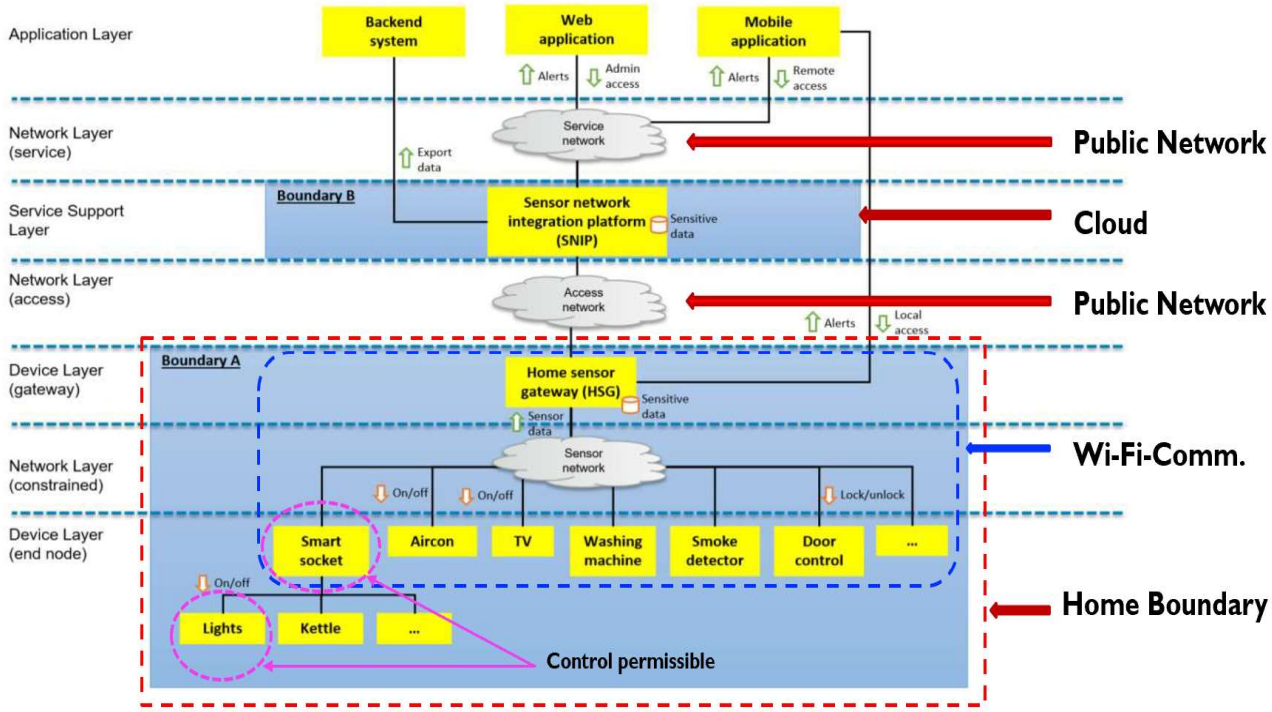


Table 12 Security Needs of Assets

Legend: H = high, M = moderate, L = low

Assets	Confidentiality	Integrity	Availability	Rationale
Sensor network integration platform (SNIP)	H	H	H	Confidentiality is high as SNIP contains sensitive data. Integrity is high to safeguard commands invocation. Availability is important because of the need to support many users.
Home sensor gateway (HSG)	H	H	M	Confidentiality and integrity is the same as SNIP. Availability is moderate as only one household is impacted.
Device: Aircon	L	M	L	Integrity is relatively more important because of monetary impact when aircon is on instead of off.
Device: TV	M	H	L	Integrity is high because compromised TV can participate in DDoS. Confidentiality is moderate because disclosure of watching habits can impact privacy.
Device: Washing machine	L	L	L	
Device: Smoke detector	L	H	M	Integrity is high, in order to gain first responder's trust in the system.
Device: Door control	M	H	H	Integrity and availability is very important for the proper operation for this device. Since this device does not contains sensitive data, confidentiality is moderate as we still want to keep activity information private as far as possible
Device: Smart socket	L	H	M	Integrity is high for safety considerations. Additional restriction might apply. For example, when kettle is connected, remote access is disallowed.
Device: Lights	L	M	M	Integrity and availability is moderate as importance of lighting is contextual. For example, use of lights at night time.
Device: Kettle	L	H	L	Integrity is high for safety considerations.

Table 13 Security Needs of Data Flows

Legend: H = high, M = moderate, L = low

Data flows	Confidentiality	Integrity	Availability	Rationale
Devices → HSG	M	H	H	The impact to CIA triad for this data flow is determined using high watermark method on the security needs of devices on the same sensor network.
HSG → SNIP	H	H	H	The impact to CIA triad for this data flow is determined using high watermark method on the security needs of SNIP and HSG.
SNIP → Backend system	H	H	L	SNIP exports data for backup at backend system. Safeguarding the confidentiality and integrity of exported data is more important, relative to availability.
Web application ←→ SNIP → HSG → Devices	H	H	H	Administration of SNIP and devices requires high confidentiality and integrity. Alerts requires high availability.
Mobile application ←→ SNIP → HSG → Device	H	H	H	Remote access to devices, through SNIP requires high confidentiality and integrity. Alerts requires high availability.
Mobile application ←→ HSG → Devices	M	H	M	Local access to devices, through HSG requires high integrity for safety considerations. Confidentiality is moderate as this data flow is transactional and not sensitive. Alerts requires moderate availability.

System Considerations:

1. Web or mobile platform for:
   a) Viewing information / alerts generated from the smart appliances.
   b) Controlling appliances (locally/ remotely/ automatically) for convenience.
2. Logging of home sensing and control events.

System specifications:

1. Smart appliances and smart socket.
2. Home Sensor Gateway.
3. Wi-Fi communication between HSG and smart appliances/smart socket, smartphone if in range. Wi-Fi to be considered to have WPA-2 protection.
4. SNIP provides interface between:
   a) HSG
   b) Backend (logging)
   c) Web application
   d) Mobile application (remote)
   (SNIP is considered to be hosted in a Linux based Cloud)

Additional system specifications assumed:

- Home Sensor Gateway, Smart TV and SNIP are TLS 1.3 capable.

Basic Privacy Considerations:

Due to under-specification of the collected data it is assumed to contain PII (e.g. home occupancy time) and *no specific privacy recommendations* are given.

Additional Security Considerations:

- As sensing data contain PII, Confidentiality is primarily important. Integrity and Availability is most important in safety and security related alerts e.g. smoke detector and door control.
- Control data to the smart appliances primarily requires Integrity and Availability.

Security Recommendations:

The security controls should be chosen based on the risk analysis and after careful considerations of the impact. The table below consists of sample recommendations and the user/implementor can choose differently according to the risks, likelihood of impact and limitations of the IoT system.

Sl. No	Devices/Infrastructure (Critical Activity ID)	Security Requirement	Control Measures	Rationale
1.	**Smoke Detector {SA, CE}**	EA – High	Challenge Response Authentication (using SHA3-256).	Integrity is High so EAUT is high. But for hash function used in challenge response authentication, only pre-image resistance property is needed, hence SHA3-256 is recommended.
		SKE - Low	NIL	Data has low confidentiality requirements.
		KM - High	Simple Password Exponential Key Exchange (SPEKE). Storage: TPM	SPEKE prevents man-in-the-middle attack. It is designed based on Diffie-Hellman and can also be realized by using Elliptic-curve cryptography.
		DI - High	HMAC with SHA3-256, 256 bits key length	Integrity is High so INT is high.

		SI - High	TPM	Integrity is High so HARD is high.
		PS - Low	NIL	Not too interesting data with the physical boundary of a home.
		AC - High	Elimination of unnecessary ports, services, protocols.	Simple device and single user system so access control measures may be limited to elimination of unnecessary ports, services, protocols although the level is high for integrity requirements.
		RDN-Medium	Multiple smoke detectors.	Availability requirement is medium.
2.	**Smart TV {SA, CE} (Export information e.g. browsing, viewing for archival purposes)**	EA - High	TLS 1.3 (X.509 certificates Curve P-521, SHA3-512, public CA)	TLS 1.3 is the current standard for transport layer security satisfying all the high requirements. The chosen TLS 1.3 cipher suite ensures high confidentiality and integrity protection by
		KM - High	TLS 1.3 (ECDHE P-521). Public Root CA. Key Storage – TEE	

		SKE- High	TLS 1.3 (AES-256 - GCM)	authenticated encryption. If TLS 1.3 is not available on the Smart TV, the security design of the IoT system should adopt mitigating measures to control of risk of exposure of highly sensitive data on the Smart TV.
		DI - High	TLS 1.3 (AES-256 - GCM)	
		SI - High	Trusted execution environment (TEE)	TEE can separate less trustworthy apps from the sensitive usage logging.
		PS - High	Simple tamper resistant setup.	High integrity requirement but the device is present within a home boundary.
		AC - High	Attribute based access control. Elimination of unnecessary ports, services, protocols.	Harden the device to reduce risk of becoming compromised.
		RDN – Low	Buffer and local storage.	Local usage information can be buffered if the SNIP is not reachable.

3.	**Aircon {SA, CE}**	EA - Low	Challenge Response Authentication (using SHA-256).	Usage of symmetric key as Aircon is only communicating with HSG.
		KM - Low	Pre-shared keys. Key storage: Non-volatile memory.	Usage of symmetric key as Aircon is only communicating with HSG.
		SKE - Low	NIL	The communication is confined to the local home network and being in the vicinity allows other methods of obtaining temperature data.
		PS - Low	NIL	Not too interesting data with the physical boundary of a home.
		AC - Low	Elimination of unnecessary ports, services, protocols.	Only standard hardening for low requirements.
		RDN - Low	NIL	A home aircon is not of sufficient importance to justify redundancy for availability.
		DI - Low	HMAC with SHA-256	Simple integrity protection to secure aircon control.
		SI - Low	NIL	A home aircon is not of enough

				importance to justify system integrity measures.
4.	**Washing machine**	Same as Aircon.		
5.	**Smart Socket {SA}**	EA - Low	Network address (MAC)	The communication is confined to the local home network.
		PS - Low	NIL	Not too interesting data with the physical boundary of a home.
		AC - Low	Elimination of unnecessary ports, services, protocols.	Only standard hardening for low requirements.
		RDN - Low	NIL	A home smart socket is most likely not of sufficient importance to justify redundancy.
6.	**Door Control {SA, CE}**	EA - High	Challenge Response Authentication (using SHA-256).	Door Control most likely has a small number of users, not justifying the overhead of public key cryptography.
		KM - High	Simple Password Exponential Key Exchange (SPEKE).	SPEKE prevents man-in-the-middle attack. It is designed based on Diffie-Hellman and

			Storage: TPM	can also be realized by using Elliptic-curve cryptography.
		SKE - Low	AES- 256 GCM	As the integrity requirement is high, AES GCM authenticated encryption requires 256 bit of security which creates higher confidentiality protection than required. AES in GCM mode of authenticated encryption is currently the most popular NIST recommended mechanism. However, lightweight standardization for authenticated encryption is in progress. Thus, it is likely to obtain better lightweight ciphers suitable for
		DI - High	AES-256 - GCM	

				IoT environments in the near future.[18]
		SI - High	TPM	Complexity of the door control is low and can be handled by a secure element.
		PS - High	Tamper resistant packaging.	To prevent unauthorized physical access.
		AC - Low	Elimination of unnecessary ports, services, protocols.	As the complexity of the door control is low a multi user operating system that provides access control is not needed.
		RDN - Low	Buffer.	Local usage information can be buffered if the SNIP is not reachable.
7.	**Home Sensor Gateway {NT, CC, CE}**	SAN - High	Behavioural monitoring and alert generation.	Monitoring the usage of the smart home devices may reveal attacks.
		AC - Medium	Identity based access control.	Medium level access control is sufficient to harden the device. Highly

18 https://csrc.nist.gov/Projects/Lightweight-Cryptography

			Elimination of unnecessary ports, services, protocols.	sensitive data is otherwise protected.
		RDN - Medium	Router operation with failover to mobile communication. (LTE/UMTS/HSPA)	Due to the system design the connection between gateway and SNIP is crucial for remote control. Consumer grade internet connections typically do not have the highest availability.
		IDP - High	Network Firewall, Anomaly based traffic analysis, Tamper-proof logs of the system status, Patch status monitoring.	Protecting the home network from connections from the internet and detecting malicious activity in the smart home.
		EA - High	TLS 1.3 (X.509 certificates Curve P-521, SHA3-512)	TLS 1.3 provides digital signature based mutual authentication, replay-resistance, forward secrecy.
		KM - High	TLS 1.3 (ECDHE P-521). Public Root CA.	High confidentiality and integrity requirements.

			Key Storage – TPM	
		SKE - High	TLS 1.3 (AES-256 - GCM)	High confidentiality requirement.
		DI - High	TLS 1.3 (AES-256 - GCM)	High integrity requirement. AES-256-GCM provides authenticated encryption.
		SI - High	TPM.	Hardware root of trust for system integrity.
8.	**SNIP {ST, CC, CE} (SNIP needs secure connectivity to backend, web and mobile application as well as HSG)**	SAN - High	Behavioural monitoring and alert generation.	Monitoring the usage of the smart home devices may reveal attacks.
		AC - High	Role based access control.	Harden the system to reduce risk of becoming compromised.
		RDN - Medium	Duplication for automatic failover and error reporting.	As single point of failure for remote control, redundancy can help to prevent downtimes.
		IDP - High	Log of authentication attempts. Tamper-proof logs of the system status (TPM). Anomaly based detection.	Attacks on the backend infrastructure require timely detection.

			Maintaining a record for all nodes connected to or disconnected from with routine updates. Patch status monitoring.	
		EA - High	TLS 1.3 (X.509 certificates Curve P-521, SHA3-512)	TLS 1.3 provides digital signature based mutual authentication, replay-resistance, forward secrecy.
		KM - High	TLS 1.3 (ECDHE P-521). Public Root CA. Key Storage – TPM	High confidentiality and integrity requirements.
		SKE - High	TLS 1.3 (AES-256 - GCM) Storage: dm-crypt with AES-256-CBC	High confidentiality requirement. Highly sensitive data requires storage encryption to ensure confidentiality if an adversary gains physical access to disks.
		DI - High	TLS 1.3 (AES-256 - GCM)	High integrity requirement. AES-256-GCM provides

				authenticated encryption.
		SI - High	TPM, Virtual TPM	Hardware root of trust for system integrity.
		PS - High	Tamper evident design.	To prevent unauthorized physical access.
		UA - High	TLS 1.3 (Elliptic Curve P-521) Two-factor authentication (X.509 certificates, biometrics, PINs etc.)	Two-factor user authentication because of high integrity requirements.

Table 14 Mapping of the Proposed Security Control Measures with IMDA Recommendations

IMDA Cyber Security Guide	Proposed ANT Architecture
Cryptographic support-CK-CS-01	SKE, ACRYPT, HASH
Cryptographic support-CK-CS-02	KM
Security function protection- CK-FP-01, CK-FP-02	SI
Identification and authentication CK-IA-01, CK-IA-02	EA, UA
Network protection-CK-NP-01	AC, IDP
Network protection-CK-NP-02	CM
Network protection- CK-NP-03, CK-NP-04	CM, IDP

Data protection- CK-DP-01	SKE, ACRYPT
Data protection- CK-DP-02	SI, CM, LCM, DI
Data protection- CK-DP-03	DI, SAN
Data protection- CK-DP-04	AC, IDP
Access protection-CK-AP-02	IDP
Access protection-CK-AP-03	AC
Access protection-CK-AP-04	PS
Access protection-CK-AP-05	UA, CM
Security management-CK-MT-01	CM
Security management-CK-MT-02	AC, CM
Security management-CK-MT-03	SI, DI, IDP
Security management-CK-MT-04	SI, DI, IDP, CM
Resiliency support-CK-RS-02	EA, EndMon
Resiliency support-CK-RS-03	IDP, RDN (partially)
Resiliency support-CK-RS-04	CM
Security Audit-CK-AU-01	EndMon, LCM
Security Audit-CK-AU-02	SI, PS, DI
Lifecycle protection- CK-LP-02, CK-LP-03	LCM
Lifecycle protection-CK-LP-04	CM, LCS
Lifecycle protection- CK-LP-05, CK-LP-06	CM
Lifecycle protection-CK-LP-07	LCM
Lifecycle protection-CK-LP-09	CM
Lifecycle protection-CK-LP-09	LCM

CONCLUSION

Geographical distribution, network heterogeneity and uncontrolled physical environment are major characteristics of Internet of Things (IoT) systems that hamper the adoption of proven traditional enterprise security architecture for the security-by-design of IoT applications. The security challenge is further exacerbated by the open and perimeterless nature of IoT devices, which would not allow any valid assumption to be made on the operating environment of IoT systems. As such, the zero trust principle becomes a key tenet in securing IoT applications. The zero trust principle adopts a posture where there is no implicit trust in any network segment, and implements the necessary security controls for identification and authentication. Data generated or processed outside secure “microperimeters” are never trusted unless explicitly authenticated and verified. Those data are only taken as a reference and not to be relied on for critical decisions. Our work aimed at designing a reference architecture framework for IoT systems based on this Open system model with security features being driven by the underlying principle of zero trust.

The Security Reference Architecture proposed in this report serves as the basis of the security-by-design concept for complex IoT systems. The design principle of this IoT security reference architecture was inspired by the three research perspectives of IoT architecture, namely Activity (semantic) perspective, Network (Internet) perspective and Things (device) perspective to understand the architecture of IoT systems, which will allow us to have an in depth understanding of the security requirements of such systems from all angles.

The Activity-centric view provided insights to key security requirements of the diversified data and processing activities inherent to IoT systems. The Network-centric analysis helped us understand the security requirements of the complex IoT communication fabric comprising of multitude of communications and networking technologies. The Things-centric view provided a bottom-up understanding of the detailed security requirements of diversified devices present in an IoT operating environment.

Risk assessment and impact analysis of IoT systems, which is an integral part of IoT Security Reference Architecture, are primarily designed around the Activity-centric view by starting with the effort of identifying critical activities, and nodes performing crucial tasks, within an IoT system. The Activity-centric understanding of security requirements is further refined by knowledge of threats and attack surfaces being identified from the Network and Things centric views of a system.

This approach to security design, based on the Activity-Network-Things (ANT) centric perspectives, enables the security architects to generate a comprehensive security requirement of IoT systems in terms of its devices, connectivity and functionality. Besides, this security reference architecture also includes mechanisms that allow IoT system designers to include organization-specific and application-specific considerations when determining security impacts of activities, hence the choice of the strength and assurance levels of the required security controls. In order to validate the effectiveness of the proposed security reference architecture, this report also included a demonstration of applying the IoT security reference architecture to three example IoT applications from the Smart Nation domain. In each of the three applications, we discussed the exercise of generating security recommendations based on the security reference architecture. The proposed architecture is shown to be flexible and effective.

Viewing IoT systems through the ANT-centric architectural approach can provide security designers and practitioners with a powerful and effective means to perform risk assessment, impact analysis and determine tailored control measures in an organized and systematic manner. The ANT-centric perspective on risk assessment and security design could thus be one of the critical success factors for designing, developing and deploying large scale IoT systems, thus allowing security-by-design of future IoT applications and achieves a considerable edge over systems with security as an afterthought.

1. Fagan, M., K. Megas, K. Scarfone, and M. Smith, *Core cybersecurity feature baseline for securable IoT devices: A starting point for IoT device manufacturers*. 2019, National Institute of Standards and Technology.
2. Boeckl, K., M. Fagan, W. Fisher, N. Lefkovitz, K.N. Megas, E. Nadeau, D.G. O'Rourke, B. Piccarreta, and K. Scarfone, *Considerations for managing Internet of Things (IoT) cybersecurity and privacy risks*. 2019: US Department of Commerce, National Institute of Standards and Technology.
3. Lin, J., W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, *A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications.* IEEE Internet of Things Journal, 2017. **4**(5): p. 1125-1142.
4. Samie, F., L. Bauer, and J. Henkel. *IoT technologies for embedded computing: A survey*. in *2016 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS)*. 2016. IEEE.
5. Gubbi, J., R. Buyya, S. Marusic, and M. Palaniswami, *Internet of Things (IoT): A vision, architectural elements, and future directions.* Future generation computer systems, 2013. **29**(7): p. 1645-1660.
6. Al-Fuqaha, A., M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, *Internet of things: A survey on enabling technologies, protocols, and applications.* IEEE communications surveys & tutorials, 2015. **17**(4): p. 2347-2376.
7. Scott Rose, Oliver Borchert, Stu Mitchell, and S. Connelly, *Zero Trust Architecture.*
8. Da Xu, L., W. He, and S. Li, *Internet of Things in Industries: A Survey. IEEE Transactions on Industrial Informatics*. DOI.
9. Burg, A., A. Chattopadhyay, and K.-Y. Lam, *Wireless communication and security issues for cyber–physical systems and the Internet-of-Things.* Proceedings of the IEEE, 2017. **106**(1): p. 38-60.
10. Medina, C.A., M.R. Perez, and L.C. Trujillo. *IoT paradigm into the smart city vision: a survey*. in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2017. IEEE.
11. Ammar, M., G. Russello, and B. Crispo, *Internet of Things: A survey on the security of IoT frameworks.* Journal of Information Security and Applications, 2018. **38**: p. 8-27.
12. Bassi, A., M. Bauer, M. Fiedler, and R.v. Kranenburg, *Enabling things to talk*. 2013: Springer-Verlag GmbH.

13. Sowe, S.K., T. Kimata, M. Dong, and K. Zettsu. *Managing heterogeneous sensor data on a big data platform: IoT services for data-intensive science*. in *2014 IEEE 38th International Computer Software and Applications Conference Workshops*. 2014. IEEE.
14. Jiang, L., L. Da Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, *An IoT-oriented data storage framework in cloud computing platform.* IEEE Transactions on Industrial Informatics, 2014. **10**(2): p. 1443-1451.
15. Shi, W., J. Cao, Q. Zhang, Y. Li, and L. Xu, *Edge computing: Vision and challenges.* IEEE internet of things journal, 2016. **3**(5): p. 637-646.
16. Stergiou, C., K.E. Psannis, B.-G. Kim, and B. Gupta, *Secure integration of IoT and cloud computing.* Future Generation Computer Systems, 2018. **78**: p. 964-975.
17. Tan, J. and S.G. Koo. *A survey of technologies in internet of things*. in *2014 IEEE International Conference on Distributed Computing in Sensor Systems*. 2014. IEEE.
18. Bormann, C., A.P. Castellani, and Z. Shelby, *Coap: An application protocol for billions of tiny internet nodes.* IEEE Internet Computing, 2012. **16**(2): p. 62-67.
19. Alliance, A., *Embedded Hardware Security for IoT Applications.* Smart Card Alliance: Princeton Junction, NJ, USA, 2016.
20. Division, C.S., I.T. Laboratory, and N.I.o.S.a. Technology, *Standards for security categorization of federal information and information systems.* NIST FIPS, 2004. **199**.
21. Grassi, P.A., M.E. Garcia, and J.L. Fenton, *Special Publication 800-63-3.* Digital Identity Guidelines, 2017.
22. NIST, *Framework for Improving Critical Infrastructure Cybersecurity*.
23. Mahmoud, R., T. Yousuf, F. Aloul, and I. Zualkernan. *Internet of things (IoT) security: Current status, challenges and prospective measures*. in *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*. 2015. IEEE.
24. Vallois, V., F. Guenane, and A. Mehaoua. *Reference Architectures for Security-by-Design IoT: Comparative Study*. in *2019 Fifth Conference on Mobile and Secure Services (MobiSecServ)*. 2019. IEEE.
25. Guth, J., U. Breitenbücher, M. Falkenthal, F. Leymann, and L. Reinfurt. *Comparison of IoT platform architectures: A field study based on a reference architecture*. in *2016 Cloudification of the Internet of Things (CIoT)*. 2016. IEEE.
26. Katie Boeckl, M.F., William Fisher, Naomi Lefkovitz, Katerina N. Megas, Ellen Nadeau, Danna Gabel O'Rourke, Ben Piccarreta, Karen Scarfone, *Draft NISTIR 8228 :*

Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks. 2018.

27. Da Xu, L., *Enterprise systems: state-of-the-art and future trends.* IEEE Transactions on Industrial Informatics, 2011. **7**(4): p. 630-640.
28. Wu, J. and W. Zhao, *Design and realization of winternet: From net of things to internet of things.* ACM Transactions on Cyber-Physical Systems, 2016. **1**(1): p. 1-12.
29. Baronti, P., P. Pillai, V.W. Chook, S. Chessa, A. Gotta, and Y.F. Hu, *Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and ZigBee standards.* Computer communications, 2007. **30**(7): p. 1655-1695.
30. Maes, R., A. Van Herrewege, and I. Verbauwhede. *PUFKY: A fully functional PUF-based cryptographic key generator*. in *International Workshop on Cryptographic Hardware and Embedded Systems*. 2012. Springer.
31. Suh, G.E. and S. Devadas. *Physical unclonable functions for device authentication and secret key generation*. in *2007 44th ACM/IEEE Design Automation Conference*. 2007. IEEE.
32. Delvaux, J., D. Gu, D. Schellekens, and I. Verbauwhede, *Helper data algorithms for PUF-based key generation: Overview and analysis.* IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2014. **34**(6): p. 889-902.
33. Yu, M.-D. and S. Devadas. *Secure and robust error correction for physical unclonable functions*. IEEE Design & Test of Computers 2010 [cited 27 1]; 48-65].
34. Zhang, T. and R.B. Lee, *Monitoring and attestation of virtual machine security health in cloud computing.* IEEE Micro, 2016. **36**(5): p. 28-37.
35. Hogan, M., B. Piccarreta, and I.I.C.S.W. Group, *Interagency report on status of international cybersecurity standardization for the Internet of Things (IoT)*. 2018, National Institute of Standards and Technology.
36. Maene, P., J. Götzfried, R. De Clercq, T. Müller, F. Freiling, and I. Verbauwhede, *Hardware-based trusted computing architectures for isolation and attestation.* IEEE Transactions on Computers, 2017. **67**(3): p. 361-374.
37. Microsoft. 2018; Available from: https://docs.microsoft.com/en-us/windows/security/information-protection/tpm/tpm-recommendations.
38. McKay, K., L. Bassham, M.S. Turan, and N. Mouha, *Report on lightweight cryptography (NISTIR8114).* National Institute of Standards and Technology (NIST), 2017.

39. Leccese, F., M. Cagnetti, and D. Trinca, *A smart city application: A fully controlled street lighting isle based on Raspberry-Pi card, a ZigBee sensor network and WiMAX.* Sensors, 2014. **14**(12): p. 24408-24424.
40. Ma, Y., M. Richards, M. Ghanem, Y. Guo, and J. Hassard, *Air pollution monitoring and mining based on sensor grid in London.* Sensors, 2008. **8**(6): p. 3601-3623.
41. Wang, P., A. Ali, and W. Kelly. *Data security and threat modeling for smart city infrastructure*. in *2015 International Conference on Cyber Security of Smart Cities, Industrial Control System and Communications (SSIC)*. 2015. IEEE.
42. Pacheco, J. and S. Hariri. *IoT security framework for smart cyber infrastructures*. in *2016 IEEE 1st International Workshops on Foundations and Applications of Self* Systems (FAS* W)*. 2016. IEEE.
43. Bačić, Ž., T. Jogun, and I. Majić, *Integrated sensor systems for smart cities.* Tehnički vjesnik, 2018. **25**(1): p. 277-284.
44. Gopi, C. and V. Lalu, *Sensor network infrastructure for AMI in smart grid.* Procedia Technology, 2016. **24**: p. 854-863.
45. Kabalci, Y., *A survey on smart metering and smart grid communication.* Renewable and Sustainable Energy Reviews, 2016. **57**: p. 302-318.
46. Energy, U.S.D.o. *Advanced metering infrastructure and Customer Systems*. 2016 [cited 2020 17/01/2020]; Available from: https://www.energy.gov/sites/prod/files/2016/12/f34/AMI%20Summary%20Report_09-26-16.pdf.
47. Geneiatakis, D., I. Kounelis, R. Neisse, I. Nai-Fovino, G. Steri, and G. Baldini. *Security and privacy issues for an IoT based smart home*. in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2017. IEEE.
48. Sridhar, S., A. Hahn, and M. Govindarasu, *Cyber–physical system security for the electric power grid.* Proceedings of the IEEE, 2011. **100**(1): p. 210-224.