Modelling of Resource-aware Information Exchange for Resource Constraint IoT Devices

Marten Fischer, Ralf Tönjes
University of Applied Sciences Osnabrück
Osnabrück, Germany
{m.fischer,r.toenjes}@hs-osnabrueck.de

***Abstract*—The Internet of Things (IoT) is the enabler for new innovations in several domains. It allows the connection of digital services with real, physical entities. These entities are devices of different categories and range in size from large machinery to tiny sensors. In the latter case, devices are typically characterized by limited resources in terms of computational power, available memory and sometimes limited power supply. As a consequence, the use of security algorithms requires expert knowledge in order for them to work within the limited resources. That means to find a suitable configuration for the algorithms to perform properly on the device. On the other side, there is the desire to protect valuable assets as strong as possible. Usually, security goals are captured in security policies, but they do not consider resource availability on the involved device and their consumption while executing security algorithms. This paper presents a resource aware information exchange model and a generation tool that uses high-level security policies as input. The model forms the conceptual basis for an automated security configuration recommendation system.**

I. Introduction

In the Internet of Things (IoT), small devices provide real-world data to applications and services in the virtual world. Protecting this data means to apply some form of security algorithm, usually even a combination of multiple algorithms, depending on the security goals. The execution of such algorithm requires resources on the IoT device, depending on the algorithm and its configuration. As a consequence, inappropriate configurations can have a negative influence on the runtime of, especially battery-powered IoT systems.

A number of security measures to protect IoT data are available. However, selecting and configuring them is not trivial. First, the assets and the necessary security goals need to be identified through talks with the stakeholder and recorded in Security Policies (SPs). The purpose of SPs is to communicate security goals among stakeholders and developers in an easy and understandable way. They should be written in a brief, but at the same time precise manner. However, at the point of creating a SP, no information about the involved IoT devices is present nor its available resources and capabilities are considered in relation to the overhead of applying the security measures.

This paper addresses the problem by a) defining a resource-aware Information Exchange Model (IEM) meta-model to describe the transport of data in a typical IoT scenario. Then a high-level Security Policy Language (SPL) is extended to describe the information exchange between the stakeholders in an abstract and easy understandable way. We show how these policies can be used to derive individual information flows of the IEM, which combine security goals and resource constraints at the same time. Finally, we discuss how the IEM can be used to find optimal security configurations. The main contributions of the paper are:

- We define a meta-model to capture message flows, its security goals, and available resources and capabilities of the participants, e.g. (IoT), at the same time.
- We extend an existing SPL and define transformation rules into the former mentioned IEM.
- We explain the transformation of SPs into an instance of the IEM in an environmental monitoring system, including different grouping modes of data assets.

The rest of this paper is organized as follows: section II addresses the topics of security policies and security measures in the IoT domain. In section III the model for a secure, resource-aware information exchange is introduced followed by a set of instructions to transform high-level policies into an instance of the IEM model in section IV. Section V presents an exemplary transformation of SP into information flows with different data assets and security goals.

II. Related Work

Applying appropriate security measures in the IoT domain is a challenging task due to the resource constraints [1]. The authors of [2]–[4] compared the resource consumption of cryptographic algorithms for both, software implementations and hardware accelerators. They show that even small priced Micro-Controller Units (MCUs) can provide satisfactory security in scenarios such as the environmental monitoring system. They also indicate that the resource consumption depends on the parametrization and that hardware implementations are not always the fastest solutions.

Other researcher approaches try to avoid large resource consumption on the IoT device. One proposal is to use lightweight security algorithms, also known as Light Weight Cryptography (LWC), on constrained devices. Dhanda et al. [5] discussed and compared the applicability of different primitives. They compared a total of 54 implementations in the classes block cipher, stream cipher, hash function and Elliptic Curve Cryptography (ECC). To compare the "lightweightness" they used, among others, the chip area of the algorithm as a metric, i.e. how many logic gates are required to implement

the algorithm in hardware. With the number of gates required, the chip area can be calculated in dependence of the internal feature size of the CMOS technology. The chip area is expressed as Gate Equivalence (GE) and directly proportional to the energy consumption. Another survey was conducted by the authors of [6], but is limited to a comparison between symmetric and asymmetric encryption algorithms. Gunathilake et al. [7] differentiate between Ultra LWC and Ubiquitous LWC, where the former applies only to specific areas, e.g. selective microcontrollers, and the latter is applicable to a wider range of platforms. In contrast to other works, they included hardware crypto accelerators into their study. An example for Ultra LWC is given by [8], where the Advanced Encryption Standard (AES) algorithm is optimized for Long-Range Wide Area Network (LoRaWAN) communication. Batina [9] studied the energy consumption of various AES implementations and showed that significant differences may exist, depending on the configuration of the security algorithms.

Another proposal to handle the resource constraints of IoT devices is to outsource (parts of) the security algorithms to non-constrained devices. The authors in [10] proposed a "Security Agent" in an edge-computing scenario to outsource complex cryptographic algorithms. A survey of edge-computing based security designs for the IoT is given by Sha [11]. Safa et al. [12] investigated the positive effects of fog and cloud computing on the security of IoT systems and presented a decision-making model for selecting the best fog nodes based on the available resources at the fog node. As such, it can be seen as resource-aware routing protocol for data through fog/edge nodes with integrated load balancer. However, they have not investigated the effects of different security algorithms and their configuration on the IoT device itself. Green et al. [13] introduced an entity named "proxy" that transforms Attribte-based Encryption (ABE) cipher-texts into less complex El-Gamal [14] style cipher-texts to be decrypted on constrained mobile devices. Manzoor et al. [15] combined Proxy Re-Encryption (PRE) algorithms with blockchain technology to implement a secure IoT data trading system. In [16], the authors propose a PRE based encryption scheme to reduce the computational costs in fog/edge nodes caused due to the offloading of away from the IoT device.

III. Information Exchange Model

This section presents the developed model to describe the information exchange between stakeholders, including the measure to secure the exchanged data in combination with the consequent additional resource consumption. The model is divided into the four concepts *Information Flow*, *Devices and Resources*, *Performance* and *Security Policy Specification*. The following subsections III-A to III-D describe each concept with focus on the aspects that influence the resource consumption caused by executing security mechanisms.

A. Information Flow $\mathcal{F}$

An Information Flow $\mathcal{F} = (\{A^*\}, M, P_P, P_R, P^*, L, \mathcal{P})$ describes the exchange of an information in the form of a message M between at least two participants , a data provider P_P and a receiver P_R. In between, other participants P^* may assist in the transport by forwarding M. Participants are connected to one another by a set of communication links L. M contains at least one asset A, which must be protected according to the required security services as stated in the security policy $\mathcal{P}$. In the context of information flows, an asset is a piece of information, e.g. sensor data. In the following, the individual components of an information flow $\mathcal{F}$ are explained in detail.

1) Asset: In the context of secure information exchange, we define an asset as digital information that can be transferred between the participants of an information flow. The asset is the entity that needs to be protected and which is of interest to an attacker. The loss/disclosure of the asset has negative effects for at least one stakeholder, e.g. loss of money and/or reputation or the revealing of confidential information.

With respect to the resource consumption in $\mathcal{F}$, an asset A can be characterized by two properties: the frequency A_f of its collection and its size. The frequency describes how often A needs to be sent, which depends on the report-strategy implemented in the use-case. Three principle strategies can be identified: a) *Pull* the data is requested by a consumer; b) *Push* the data is sent by the IoT device, if needed or not; and c) *Event-based* data is sent only when its value changes. In an IoT scenario, the size of an asset can vary between a few bits for numerical sensor readings to several megabytes for high-quality video streams.

2) Message: The message is used to transfer one or more assets within an information flow $\mathcal{F}$ using a specific communication protocol (including protocol overhead). As mentioned before, a message M is transferred from one participant (a data provider P_P) to at least one other participant (a receiver P_R), but may be routed through/processed by, any number of participants (P^*) in between. The frequency to send a message M_f may be determined by the measurement frequency of the asset A_f. In the simplest case, all assets are measured at the same time and transferred in one message. In this case, both frequencies are the same ($M_f = A_f$).

3) Participant: Participants are any number of entities involved in $\mathcal{F}$. Two participants are mandatory: a *Provider* (P_P) that transmits M to a *Receiver* (P_R). During transfer, M may pass additional entities, which can be classified as active or passive. Passive participants will be called *Gateway* and active participants *Proxy*. With regard to security mechanisms, the difference between both classes is that gateways just forward a message to the next participant, while proxies employ additional security mechanisms. The name proxy was chosen to relate to the PRE [17] schemes.

4) Communication Link: In an information flow $\mathcal{F}$, each participant is connected to the next one through a Communication Link. The Participants may support different communication technologies, such as Wi-Fi, and establish multiple links, e.g. to separate signalling and payload messages. The Communication Technology influences the resource consumption while sending or receiving M.

B. Devices and Resources concept

The second concept of the IEM describes the devices and their resources available to receive or transmit messages M of the participants involved in a flow $\mathcal{F}$. Each device has a set of Resources (and Capabilities) that can be used in $\mathcal{F}$. In the context of secure message exchange in a flow $\mathcal{F}$ using resource constrained IoT devices the most relevant resources are: computational power, memory, the available communication technologies, the general energy consumption and as non-technical metric the financial cost. Capabilities include hardware accelerators for cryptographic algorithms. In addition to the resources provided by the device, the use-case itself provides resources. In the context of an information exchange, these are typically described in the form of use-case requirements and include: real-time requirements, the available energy source and the financial budget.

C. Performance concept

A *Performance Metric (P_M)* indicates the ability of a device to perform a security algorithm and the resulting resource consumption. P_M can be determined and saved into a knowledge-base by either analysing the security algorithm itself or by experimentally executing the algorithm on the target device. Alternatively, the execution of the algorithms can be emulated in order to evaluate that the avail resources are sufficient. There are two types: a P_M either describes the amount of resources that are going to be consumed when executing the algorithm, or it describes the amount of data (assets) that can be processed in time, e.g. how many bytes of data can be encrypted/decrypted per second.

In addition, we define *Performance Indicators (P_Is)* as a combination of all relevant P_M that influence an algorithm for a specific use-case. The P_M can be weighted to reflect use-case resources that describe a hard threshold, e.g. real-time requirements. An ideal solution would be, if the P_I could be a normalized value, ranking suitable algorithms in terms of resource consumption on the one hand and the security level on the other.

D. Security Specification and Configuration concept

To capture all security requirements, a system designer consults with the stakeholder. At this stage, a security policy is described at high-level. Later on, the policies are refined and concretized into lower levels. To reflect this, the security specification in the IEM is also done at different levels, which are explained in the following. The terminology used follows the definition of [18].

1) Security Service (S_S): is the most abstract specification. Here, together with stakeholders, security goals are formulated. For example, a stakeholder may specify that the information exchange shall be *Confidential* or that the asset(s) shall be protected from manipulation (*Integrity*).

2) Security Mechanism (S_M): is a mechanism to map a S_S. There may be multiple mechanisms to implement one S_S. For example, the S_S Confidentiality can be achieved using the S_Ms *Encryption* or *Steganography*. The security mechanisms *Digital Signature* or *Message Authentication Code (MAC)* can be used to achieve the S_S *Integrity*.

3) Security Implementation (S_I): name specific algorithms to implement a S_M, a set of parameters for the algorithm and also references additional S_I, if necessary. For example, the S_M *Encryption* can be implemented with symmetric (e.g. AES, Twofish, Blowfish) or asymmetric (e.g. RSA, ABE) algorithms. The reference to any additional S_I is required, because implementing a S_M may require multiple algorithms, e.g. *Digital Signatures* use an asymmetric encryption algorithm and a hash function to "reduce" the amount of data to be signed.

4) Security Configuration (S_C): is a set of S_I and their configurations, that are usable in an information flow, that is, they are within the boundaries of the available resources of the IoT devices. Example configuration parameters for the S_M encryption include key length, padding mechanism, specific curves in ECC, and for block ciphers the mode of operation.

IV. High-level Security Policy Language (HSPL) to IEM Transformation

This section describes the transformation from a SP to an instance of the IEM. We decide to use and extend the High-level Security Policy Language (HSPL) developed in the projects SECURED [19] and ANASTACIA [20]. ANASTACIA investigated ways to define security requirements for Software Defined Networks (SDN). They developed a SPL, named High-level Security Policy Language (HSPL), to be used by typically non-technical end-users and is based on a set of predefined high level syntax. The syntax follows a subject - predicate/action - object scheme, followed by the possibilities to add further conditions and restrictions in the form of key-value pairs. The SPL can be extended by re-defining or adding own expressions.

After the security requirements have been recorded within the high-level SPs, the individual message flows need to be extracted. For this, the policies are transformed into an IEM instance as introduced in section III. The overall approach is as follows: every time a certain action type is identified, a new flow $\mathcal{F}$ is assumed. Policies using the same HSPL-object are assumed to belong to the same flow. This way, by iterating over all policies, the participants can be added to the flows. At the end, possibilities to merge flows together are searched. That is the case when all participants and their order are identical in different flows. A practical example for such two flows could be a temperature/humidity sensor, which provides both measurements at the same time. The following subsection give details about the different steps.

A. Element Identification

In a first step of the transformation, the elements, as described in sections III, need to be identified. More specifically, the Participants and their roles as well as the asset(s) must be found within the HSPL policies. To identify an asset A, we define a special type of action named *protective*. A HSPL-object affected by a protective action is classified as an

TABLE I
PARTICIPANT IDENTIFICATION BASED ON HSPL ACTIONS

Type → Action ↓	Provider	Gateway	Receiver	Proxy	Asset
provides	✓	-	-	-	-
publish	✓	-	-	-	-
protects	✓	-	-	-	✓
(not)authorised to access	-	-	✓	-	(✓)
receives	-	-	✓	-	-
subscribes	-	-	✓	-	-
requests	-	-	✓	-	-
forwards	-	✓	-	-	-
converts	-	-	-	✓	(✓)

asset. We defined four actions as protective actions: protects, converts, (not) authorized to access. While the "protect" action is trivial, the others imply the security requirements indirectly. For example, the "not authorised to access" action indicates that the following object was protected at the provider's side.

In order to identify the participants of a flow within HSPL policies, we introduce a mapping from HSPL-subjects to the participants of an information flow through a set of predefined HSPL-actions. In other words, we predefine the available actions for a subject depending on its later role. This is in contrast to the original HSPL definition by the ANASTACIA project, where in theory each subject could execute each action. Table I shows the mappings between the HSPL-actions and the participants. A special meaning has the "requests" action of a receiver, as it indicates that the report-strategy Pull is to be used. In addition, the HSPL-object in a policy using one of the predefined actions can be interpreted as an asset in an information flow, as shown in the column far to the right of the table.

B. Grouping Modes

After the participants and their roles have been identified, the question of how to apply the security measure S_M needs to be addressed. Different strategies can be considered, relevant for the consumption of resources. For example, if the flow $\mathcal{F}$ contains multiple assets, but all are treated the same, these assets can be grouped before the security measure is applied. This may be beneficiary with regard to the resource consumption, since certain operations don't need to be executed multiple time, e.g. initialising cryptographic routines or padding the data for block ciphers. Different grouping modes are possible. The grouping modes are: no grouping (each S_S is applied to each asset separately); grouping by S_S (all S_S that can be applied to one or more assets are grouped); grouping by asset (all assets requiring the same S_S are grouped); .

C. Resource Annotation

Although not explicitly stated by the ANASTACIA project [21], [22], the examples for HSPL-fields as provided in the documentation indicate that they only allow to specify characteristics and limitations for HSPL-objects. To be able to annotate the HSPL policies with additional information required in the IEM, we introduce a set of specific HSPL-fields that affect the HSPL-subject as well. In that respect, we extend the original concept with the following HSPL-fields to further describe HSPL-subjects.

is a: This field key is used to specify a device type for the corresponding HSPL subject.

every: This field key is used to specify the update frequency of the IoT device in which new data is provided. The field value has to be some sort of time value, such as 1 second(s).

within: With this field key, real-time requirements in the provision of data can be specified. In contrast to **every**, this field describes the maximum transfer time of the message M from provider Participant Provider (P_P) to receiver Participant Receiver (P_R).

has energy source: This field key is important for IoT devices with a limited energy source, i.e. that are battery powered. Its value describes the amount of energy stored within the battery in mAh (mill-amp-hour).

has size: With this field, the size of an asset (i.e. the object) can be specified.

with topic: This field describes the message's topic in a publish/subscribe pattern. The topic's size has to be added to the size of the message.

V. IMPLEMENTATION

For the previously presented concept, we implemented a transformation tool applying the transformation rules, as presented in section IV, from the high-level SPs into an instance of the IEM. To visualize the information flows in an IEM, we choose to use Business Process and Modelling Notation (BPMN) as representation format. The use of an existing BPMN extension would benefit from recognizability and available experience. In [23], Zarour et al. provide an extensive overview over several available BPMN extensions and give a statistical evaluation about types of extensions. They show that the majority of BPMN extensions introduce new graphical elements to represent special, usually domain specific, process behaviour. Chergui provides in [24] an overview of security related BPMN extensions. Bocciarelli [25], [26] developed BPMN extensions to model task resources in the context of Cyber Physical Systems (CPS) and the industry 4.0. However, since no extension covers both aspects, security and resource limitations, at the same time, the use of a combination of two extensions is necessary.

For the IEM, the modelling of the operations of the individual participants can be done in a very simplified way. As depicted in figure 1, each participant operates in an endless loop, where at least one operation is related to the transfer of a message. With the exception of a participant of type Gateway, another operation is to apply/remove the security measure. Between loop iterations, the participant of type Receiver delays for a specific amount of time to match a message's frequency M_f. Meanwhile, the other participants start their operations upon reception of message M.

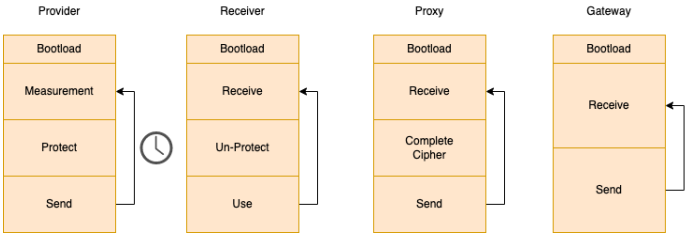

Fig. 1. Participant operations

The transformation is illustrated on an example with a sensor providing temperature, humidity and air pressure readings (e.g. the BME280[1]). The S_Ss for the temperature and humidity shall be *confidentiality* and *integrity*. The S_S for the asset "air pressure" is only *integrity*. Within the information flow, the message shall pass through a gateway (Access Point). For the purpose of better visibility, the tasks to read out the sensor and use this data, have been merged into a single subprocess. This results in the following 9 HSPL policies:

1) Sensor; protects Confidentiality, Integrity; temperature; has size $\rightarrow$ 16 bytes; every $\rightarrow$ 1s
2) Sensor; protects Confidentiality, Integrity; humidity
3) Sensor; protects Integrity; air pressure
4) Bob; authorized to access; temperature
5) Bob; authorized to access; humidity
6) Bob; authorized to access; air pressure
7) Access Point; forwards; temperature
8) Access Point; forwards; humidity
9) Access Point; forwards; air pressure

Figure 2 shows the BPMN representation of the resulting information flow with grouping mode "grouping by asset". As all assets have the same P_P, P_R, and the gateway, only one single flow is generated. The participants P_P and P_R (top and bottom) contain four tasks each, two of which to apply the security services. The security tasks are depicted with a padlock symbol, as suggested by [27]. Furthermore, we also acknowledge the need to "undo" or validate a security measure at the receiver side, which will be represented with an open padlock. The S_S applied in the task is illustrated with orange symbols. Here, we use the symbols defined in SecBPMN [28]. That is a symbol with two hands shaking for *confidentiality* and a symbol with a white document for *integrity*.

The sequence flows (chain of tasks within a BPMN pool) relate with the operations for the different types of participant, as introduced in figure 1. On the P_P's side, it covers the gathering of the assets from the sensors. The tasks should be annotated with the resource consumption to read out the hardware sensor. Like with the BME128 sensor, this is often a single, integral step for all assets. Thus, from a resource consumption point of view, it is possible to merge all these tasks into one task/subprocess. In general, as a detailed sequence

[1]https://www.bosch-sensortec.com/products/environmental-sensors/humidity-sensors-bme280/

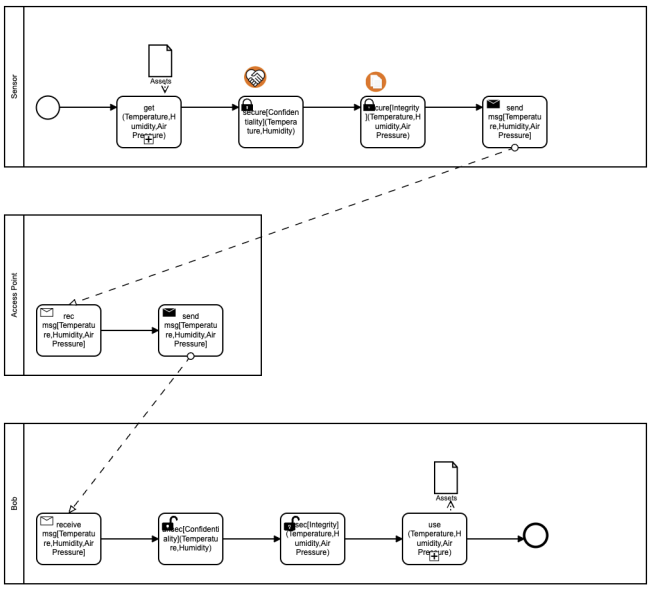

Fig. 2. Information flow represented as BPMN process

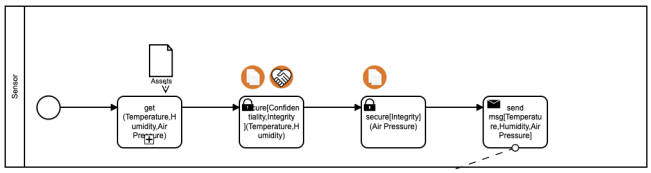

Fig. 3. Information flow with grouping by Security Service

flow is not necessary as long as the resource consumption is known, the same merging can be done on the P_R's side.

The Access Point in the middle simply receives the message and directly afterwards sends it to the P_R Bob. The resource properties are provided via an input form in a property-panel.The implementation is an extension to the bpmn-js project [29], an open source BPMN modeller written in JavaScript. The transformer tool is implemented in Swift and is still a work in progress. The GUI allows constructing new HSPL policies out of the redefined sets of HSPL elements, i.e. subjects, actions, objects and fields.

Figure 3 shows the tasks of the participant P_P when using the grouping mode "grouping by S_S". As before, two tasks are generated to protect all three assets. However, in contrast to the first example, protecting the *Integrity* in the later task only processes one asset, namely the air pressure.

VI. Conclusion and Future Work

This paper addressed the topic of applying security measure while providing data with resource constrained devices in an IoT environment. Furthermore, the paper presented an IEM capable of capturing available/required resources. As these policies are used by non-expert stakeholders, the use of a high-level SPL was proposed, together with a set of transformation rules to derive individual information flows as part of the resource-aware Information Exchange Model (IEM).

As a next step to realize the envisioned security configuration recommendation system is to determine the P_I. That means to determine relevant sets of P_M for available IoT devices. Here, an extendable emulation framework is planed, able to emulate the resource consumption of an information flow on a microcontroller. More precisely, the resource consumption during the execution of various security algorithms and configurations is in the focus. First experiments with the QEMU emulator for the ESP32 microcontroller showed mixed results with respect to the runtime when compared with the real device.

ACKNOWLEDGEMENT

This work is part of the research project "I4sec - Sichere Maschinenkommunikation und Fernwartung von Sensoren in der Produktion", funded by the Federal Ministry of Education and Research of Germany (BMBF).

REFERENCES

[1] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh, "Iot security: ongoing challenges and research opportunities," in *2014 IEEE 7th international conference on service-oriented computing and applications*. IEEE, 2014, pp. 230–234.

[2] B. Pearson, L. Luo, Y. Zhang, R. Dey, Z. Ling, M. Bassiouni, and X. Fu, "On misconception of hardware and cost in IoT security and privacy," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pp. 1–7, ISSN: 1938-1883.

[3] P. Kietzmann, L. Boeckmann, L. Lanzieri, T. C. Schmidt, and M. Wählisch, "A performance study of crypto-hardware in the low-end IoT," in *Proceedings of the 2021 International Conference on Embedded Wireless Systems and Networks*, ser. EWSN '21. Junction Publishing, pp. 79–90.

[4] P. S. Munoz, N. Tran, B. Craig, B. Dezfouli, and Y. Liu, "Analyzing the resource utilization of AES encryption on IoT devices," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1200–1207, ISSN: 2640-0103.

[5] S. S. Dhanda, B. Singh, and P. Jindal, "Lightweight cryptography: A solution to secure IoT," vol. 112, no. 3, pp. 1947–1980. [Online]. Available: https://doi.org/10.1007/s11277-020-07134-3

[6] I. K. Dutta, B. Ghosh, and M. Bayoumi, "Lightweight cryptography for internet of insecure things: A survey," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, 2019, pp. 0475–0481.

[7] N. A. Gunathilake, W. J. Buchanan, and R. Asif, "Next generation lightweight cryptography for smart iot devices: : Implementation, challenges and applications," in *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 707–710.

[8] K.-L. Tsai, Y.-L. Huang, F.-Y. Leu, I. You, Y.-L. Huang, and C.-H. Tsai, "Aes-128 based secure low power communication for lorawan iot environments," *IEEE Access*, vol. 6, pp. 45 325–45 334, 2018.

[9] L. Batina, A. Das, B. Ege, E. B. Kavun, N. Mentens, C. Paar, I. Verbauwhede, and T. Yalçın, "Dietary recommendations for lightweight block ciphers: power, energy and area analysis of recently developed architectures," in *International Workshop on Radio Frequency Identification: Security and Privacy Issues*. Springer, 2013, pp. 103–112.

[10] R.-H. Hsu, J. Lee, T. Q. S. Quek, and J.-C. Chen, "Reconfigurable security: Edge-computing-based framework for IoT," vol. 32, no. 5, pp. 92–99, conference Name: IEEE Network.

[11] K. Sha, T. A. Yang, W. Wei, and S. Davari, "A survey of edge computing-based designs for IoT security," vol. 6, no. 2, pp. 195–202. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864818303018

[12] N. S. Safa, C. Maple, M. Haghparast, T. Watson, and M. Dianati, "An opportunistic resource management model to overcome resource-constraint in the internet of things," vol. 31, no. 8, p. e5014, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpe.5014. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5014

[13] M. Green, S. Hohenberger, B. Waters *et al.*, "Outsourcing the decryption of abe ciphertexts." in *USENIX security symposium*, vol. 2011, no. 3, 2011.

[14] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[15] A. Manzoor, M. Liyanage, A. Braeke, S. S. Kanhere, and M. Ylianttila, "Blockchain based proxy re-encryption scheme for secure iot data sharing," in *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, 2019, pp. 99–103.

[16] O. A. Khashan, "Hybrid lightweight proxy re-encryption scheme for secure fog-to-things environment," *IEEE Access*, vol. 8, pp. 66 878–66 887, 2020.

[17] K. Suksomboon, A. Tagami, A. Basu, and J. Kurihara, "In-device proxy re-encryption service for information-centric networking access control," in *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, 2018, pp. 303–306.

[18] G. Patz, M. Condell, R. Krishnan, and L. Sanchez, "Multidimensional security policy management for dynamic coalitions," in *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01*, vol. 2, pp. 41–54 vol.2.

[19] M. Vallini, "Fp7 project secured deliverable "d4.1 policy specification"," April 2015.

[20] A. M. Zarca, J. B. Bernabé, J. Ortíz, and A. Skarmeta, "H2020 project anastacia deliverable "d2.5 policy-based definition and policy for orchestration final report"," December 2018.

[21] ANASTACIA Project - advanced networked agents for security and trust assessment in CPS / IOT architectures. [Online]. Available: http://www.anastacia-h2020.eu/

[22] AANASTACIA Project - GitLab repsitory. [Online]. Available: https://gitlab.com/anastacia-project

[23] K. Zarour, D. Benmerzoug, N. Guermouche, and K. Drira, "A systematic literature review on BPMN extensions," publisher: Emerald Publishing Limited. [Online]. Available: https://www.emerald.com/insight/content/doi/10.1108/BPMJ-01-2019-0040/full/html

[24] M. E. A. Chergui and S. M. Benslimane, "A valid BPMN extension for supporting security requirements based on cyber security ontology," in *Model and Data Engineering*, ser. Lecture Notes in Computer Science, E. H. Abdelwahed, L. Bellatreche, M. Golfarelli, D. Méry, and C. Ordonez, Eds. Springer International Publishing, pp. 219–232.

[25] P. Bocciarelli, A. D'Ambrogio, A. Giglio, and E. Paglia, "A BPMN extension to enable the explicit modeling of task resources."

[26] ——, "A BPMN extension for modeling cyber-physical-production-systems in the context of industry 4.0," in *2017 IEEE 14th International Conference on Networking, Sensing and Control (ICNSC)*, pp. 599–604.

[27] K. S. Sang and B. Zhou, "BPMN security extensions for healthcare process," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, pp. 2340–2345.

[28] M. Salnitri, F. Dalpiaz, and P. Giorgini, "Modeling and verifying security policies in business processes," in *Enterprise, Business-Process and Information Systems Modeling*, ser. Lecture Notes in Business Information Processing, I. Bider, K. Gaaloul, J. Krogstie, S. Nurcan, H. A. Proper, R. Schmidt, and P. Soffer, Eds. Springer, pp. 200–214.

[29] "bpmn-js - BPMN 2.0 for the web," original-date: 2014-03-10T12:57:00Z. [Online]. Available: https://github.com/bpmn-io/bpmn-js