sustainability

MDPI

Article

Hybrid Logical Security Framework for Privacy Preservation in the Green Internet of Things

Isha Batra [1], Sahil Verma [1], Arun Malik [1], Kavita [1], Uttam Ghosh [2], Joel J. P. C. Rodrigues [3,4], Gia Nhu Nguyen [5,6], A. S. M. Sanwar Hosen [7,*] and Vinayagam Mariappan [8,*]

[1] School of Computer Science and Engineering, Lovely Professional University, Phagwara 144411, India; isha.17451@lpu.co.in (I.B.); sahilverma@ieee.org (S.V.); arun.17442@lpu.co.in (A.M.); kavita@ieee.org (K.)

[2] Department of EECS, Vanderbilt University, Nashville, TN 37240, USA; uttam.ghosh@vanderbilt.edu

[3] Post-Graduation Program in Electrical Engineering, Federal University of Piauí (UFPI), Teresina 64049-550, Brazil; joeljr@ieee.org

[4] Covilhã Delegation, Instituto de Telecomunicações, 6201-001 Covilhã, Portugal

[5] Graduate School, Duy Tan University, Da Nang 550000, Vietnam; nguyengianhu@duytan.edu.vn

[6] Faculty of Information Technology, Duy Tan University, Da Nang 550000, Vietnam

[7] Division of Computer Science and Engineering, Jeonbuk National University, Jeonju 54896, Korea

[8] Advanced R&D Department, SMR Automotive Modules, Bucheon 14556, Korea

* Correspondence: sanwar@jbnu.ac.kr (A.S.M.S.H.); vinayagam@ieee.org (V.M.)

Received: 21 May 2020; Accepted: 26 June 2020; Published: 9 July 2020

check for updates

Abstract: Lately, the Internet of Things (IoT) has opened up new opportunities to business and enterprises; however, the cost of providing security and privacy best practices is preventing numerous organizations from adopting this innovation. With the proliferation of connecting devices in IoT, significant increases have been recorded in energy use, harmful contamination and e-waste. A new paradigm of green IoT is aimed at designing environmentally friendly protocols by reducing the carbon impact and promote efficient techniques for energy use. There is a consistent effort of designing distinctive security structures to address vulnerabilities and attacks. However, most of the existing schemes are not energy efficient. To bridge the gap, we propose the hybrid logical security framework (HLSF), which offers authentication and data confidentiality in IoT. HLSF uses a lightweight cryptographic mechanism for unique authentication. It enhances the level of security and provides better network functionalities using energy-efficient schemes. With extensive simulation, we compare HLSF with two existing popular security schemes, namely, constrained application protocol (CoAP) and object security architecture for IoT (OSCAR). The result shows that HLSF outperforms CoAP and OSCAR in terms of throughput with low computational, storage and energy overhead, even in the presence of attackers.

Keywords: green IoT; ICT; authentication; confidentiality; cryptography; security framework

1. Introduction

The Internet of Things (IoT) leads to a revolutionary change in the lifestyle of users [1]. Every device that is connected to the IoT works in a smart way to make the world technology-dependent [2]. IoT works in numerous applications, such as inventory, health care, and smart homes [3]. Therefore, users expect a high level of security and privacy from the IoT, requiring a security framework [4]. The inbuilt security solutions in the IoT are susceptible to attacks, such as denial of service (DoS), spoofing, and many more [5]. A security framework is evaluated by assessing its security features, such as authentication, or confidentiality in this case [6,7]. The next step for evaluating a framework is to check whether the data collected are authentic or not [8]. The last evaluation measure for a security framework is the synthesis or mining of data to make fruitful decisions. A security framework can be competent if it can meet the security requirements and can make decisions efficiently in real-time [9]. A security framework tends to provide overall system security. As shown in Figure 1 below, there are four operation layers in the IoT. These four layers—that is the perception, network, transport, and application layers—take care of the security implementation at different levels.

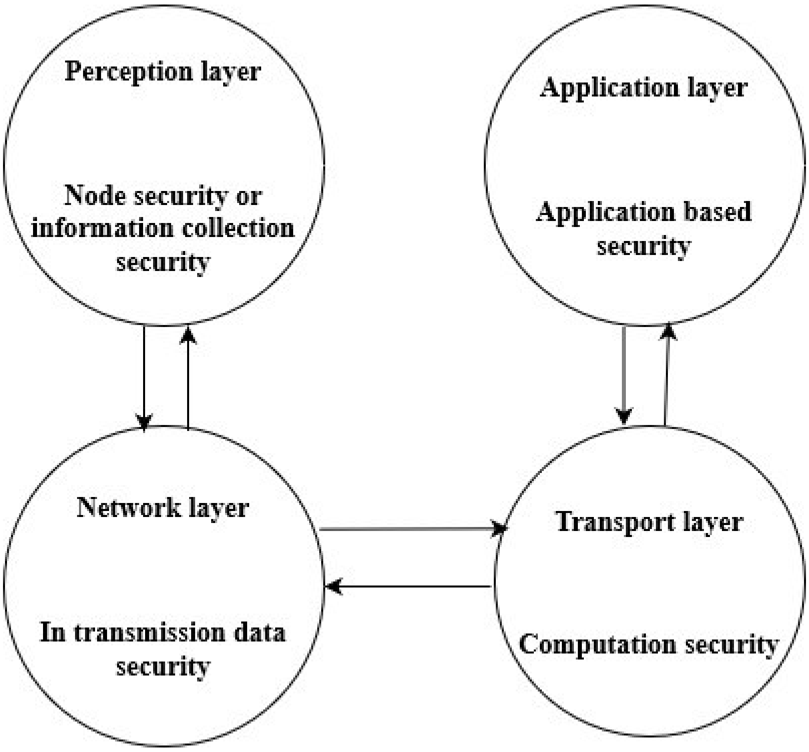
Perception layer
Node security or information collection security
Application layer
Application based security
Network layer
In transmission data security
Transport layer
Computation security

Figure 1. Security concerns at each layer of the Internet of Things (IoT).

The IoT consists of resource-constrained devices like radio frequency identification (RFID), sensors that are battery operated. Therefore, special attention should be paid to limit the use of their resources, as well as to offer security at the same time. Lightweight cryptography solutions offer both security as well as performance. The main reasons for applying lightweight cryptography in IoT are summarized below:

- End-to-end communication efficiency: when two resource-constrained devices communicate using lightweight solutions, the overall energy consumption will be reduced. Hence, the end-to-end communication will become efficient.
- Increased number of connections: as a lightweight solution requires fewer resources, any resource-constrained device can connect to the network. The number of connections in the network will thus increase.

Lightweight block ciphers work well over conventional solutions due to their lightweight design considerations, as specified below:

- Small key size: the key size selected by a lightweight block cipher should be comparatively less than those of conventional block ciphers. The National Institute of Standards and Technology (NIST) restricts the minimum key size to 112 bits. A key size less than this is more susceptible to brute force attacks.
- Small block size: the block size chosen for the lightweight cipher should be less than those of conventional ciphers. For example, if the block size is chosen as 64 bit, as compared to 128 bit,

the Advanced Encryption Standard (AES), a greater number of plaintext blocks can be encrypted. Moreover, the memory requirements will be less.

- Simple round structure: the rounds designed for lightweight ciphers should be simpler than the conventional cryptographic algorithms. For example, a round can be made simpler by replacing an 8-bit Substitution (S)-Box with a 4- bit S-Box. This will also reduce memory requirements. This may reduce the level of security, which can be improvised by increasing the total number of rounds.
- Simple key schedule: the generation function of key schedule in lightweight designs must generate the subkeys very fast. The simpler a key schedule is, the less power consumption and memory will be required by the algorithm. Using a simple key schedule may lead to attacks like a weak key, related key, or chosen key attack, but that can be overcome by using a secure and frequent function for key generation.
- Fewer implementation requirements: A device should support either encryption or decryption. Only the required operations of the cipher should be implemented rather than implementing the full cipher.
- Purpose of green IoT: green IoT focuses on reducing IoT energy usage, a necessity for satisfying the world's appetite for the maintainability of everything being intelligent and reducing CO_2 emissions. Green IoT comprises designing and leveraging aspects [10]. The design elements of green IoT refer to developing registering devices, correspondence conventions, energy efficiency, and networking architectures [11]. Leveraging the IoT element seeks to reduce or eliminate emissions of CO_2, reduce the contaminations, and enhance energy efficiency. On the other hand, the enabling technologies for green IoT are called information and communication technology (ICT) technologies [12].

1.1. Motivation

With the advancement of technology, every internet user is more inclined towards the smart activities that can be performed using IoT. With this smartness, security vulnerabilities also become an issue, when user details are processed online. This kind of call is required for a security framework for IoT environments. Security frameworks can be designed at every architectural layer of IoT. This work primarily works on the security framework on the application layer. Different frameworks for security exist in IoT, but the existing solutions use the traditional heavyweight security mechanisms like AES in various modes or asymmetric cryptographic solutions like RSA. These heavyweight solutions can take away all the resources and will decay the power sources of the devices. Therefore, this research paper proposes a security framework that uses a lightweight security alternative that will consume minimal resources and power. In this way, this system is also helpful for protecting the environment.

1.2. Contributions

Three major phases of security are taken into consideration while designing a security framework for IoT; namely, registration, authentication, and data security.

- Registration: This is the first phase every device has to go through. Whenever a device joins the network, the identity of the device is registered on the server. This is a one-time process.
- Authentication: Once the registration is done and the device is in an active state and has some information to share with the server, the device has to first authenticate its identity to the server. The server in response also authenticates itself to the device. The moment mutual authentication is over, a device can initiate the data transfer process.
- Data Security: The last but the important concern is that every time information is shared between the device and the server, the data need to be secure. This phase ensures that the data shared are not readable as well as not alterable by a third party.

Therefore, the main contribution of this work is the proposal of a security framework that uses a smaller key size, fewer rounds with an easy but tricky round structure in the process of registration, authentication as well as offering data security. On the other hand, the existing frameworks tend to use a large key size and a complex round structure. This makes the proposed framework a better, more lightweight solution as compared to the existing counterparts.

1.3. Organization

The rest of the paper is organized as follows. In the second section, the literature review is conducted that highlights the existing security frameworks in IoT by specifying their security features and modes of operation. In this section, two existing frameworks are detailed; one is constrained application protocol (CoAP), and the second is object security architecture for IoT (OSCAR). The third section proposes a new security framework, HLSF, for IoT categorized into three distinct phases, where the first is registration, the next is authentication and the final phase is data security. This section elaborates on the flow of steps that are to be followed for one-time registration, authentication, and every-time data security required during data transmission. Later, in the next section, the existing security frameworks CoAP and OSCAR are compared with the proposed security framework HLSF using the COOJA simulator, and based on result analysis, certain discussions and decisions are made. Performance parameters used for comparison are memory requirements, energy overhead, computation overhead, and communication rate. Finally, the last section concludes with the state of the art and the working efficiency of the HLSF proposed.

2. Related Work

After conducting the theoretical and analytical study, this section describes the recent state of existing security frameworks in IoT. There are several existing security frameworks in IoT, achieving the same target while following different approaches [13–15]. Each framework is designed based on the same level of expectancy, as specified below

- Software reliance in each framework to carry out the whole process
- Set of protocols required to initiate and set up the communication among the devices.
- Contribution of the security framework in maintaining the security and privacy in IoT.

Considering the expectancy from a security framework and in authentication, cryptographic solutions [16] used by the framework are the prime concern of this section. The frameworks analyzed in this study are the constrained application protocol (CoAP) framework, and object security framework for IoT (OSCAR).

2.1. Constrained Application Protocol (CoAP) Framework

CoAP was proposed by constrained restful environment working group (CORE) in the Internet Engineering Task Force (IETF) [17]. CoAP works for constrained devices at the application layer. For communication, IPV6 over Low Power Personal Area Network (6LoWPAN) provides the usage of IPv6 communication among the sensing devices. IoT devices can communicate in CoAP by using the user datagram protocol (UDP) at the transport layer and 6LoWPAN [18]. Interactions in a constrained network like IoT running CoAP can be either between devices or the client/server, where one of the devices can act as a client and one dedicated device acts as a server [19]. Figure 2 shows the integration of CoAP with the Internet. CoAP itself acts as an internal network, which means that a CoAP client request can only be processed by the CoAP server. CoAP, otherwise, can be extended and can process HTTP client requests using the CoAP/HTTP mapping process, as CoAP acts as a subset of HTTP. The 6LoWPAN border router (6LBR) can be used to establish this connection.

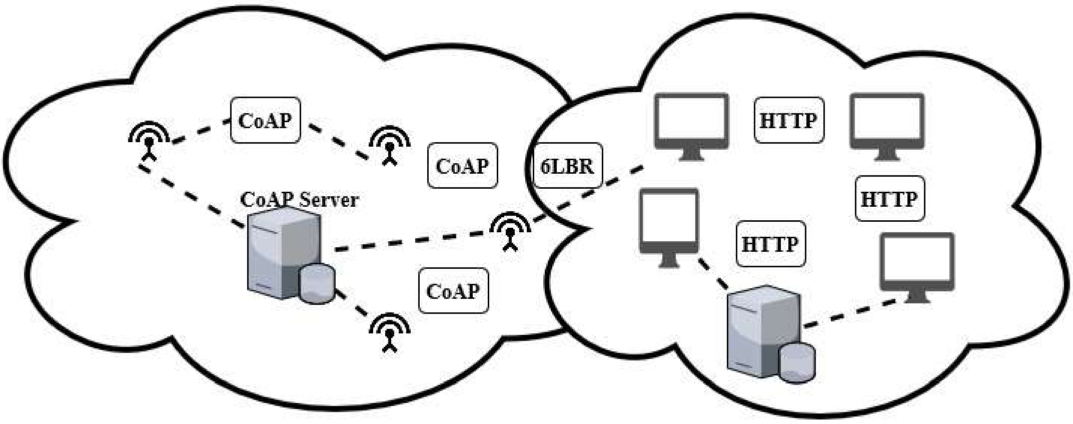

Figure 2. Constrained application protocol (CoAP) Network.

The CoAP framework uses the CoAP protocol working at the application layer, the transport layer's UDP protocol and, in the end, 6LoWPAN is used at the network layer [20]. At the network layer, 6LoWPAN is used, as it commercializes with the constrained networks like IoT. At the transport layer, UDP is used for fast transmission as compared to the reliable countermeasure transmission control protocol (TCP). This is due to the reliability mechanism offered by the message layer of CoAP. At the application layer, CoAP operates at distinguished sub layers. The responsibility of the request/response layer is to employ methods like get, put, post, and delete for accessing the resources in the CoAP network [21–23]. The number of requests and the mapping among their semantically correct responses also forms the responsibility of this layer.

CoAP offers security features by using datagram transport layer security (DTLS) over UDP instead of TCP. DTLS is designed to offer end-to-end security. As it runs with UDP, it can be used in numerous constrained applications like voice over IP (VoIP), real-time communication. DTLS in CoAP offers security features such as authentication, confidentiality, integrity, key sharing mechanism [24]. This research work emphasizes the authentication and the confidentiality security aspect of CoAP. CoAP comes in four security options, namely NoSec, PreShared Key, Certificates, and RawPublic key.

2.2. Object Security Framework for IoT (OSCAR)

The OSCAR framework works on the consumer–producer model. In IoT, consumers are the end devices, such as accessories used by humans, and the actuating devices that intake data from the producers, such as sensors, smartwatches, smart meters, and motion sensors. The responsibility of offering security in OSCAR lies with the producers. Security is provided to the data at rest or during transmission. The major emphasis of OSCAR is on the object/device security. Global applications, such as smart cities, work on OSCAR, as several consumers/clients are requesting the services of constrained servers [25]. Figure 3 represents the OSCAR framework as the typical consumer–producer model.

OSCAR needs authorization servers to restrict the access of resources by the consumers. For authentication, OSCAR uses the simple concept of elliptic curve digital signatures. A cryptographic security solution to offer confidentiality is provided by using AES in CCM mode is used.

There are numerous other security add-ons that are frequently made in the literature for IoT. With the increasing number of sensors required for IoT surveillance, there comes a requirement of large-scale sensor-based designs for operating these systems. For supporting these applications which are based on real-time with minimum delay, routers or switches are visualized [26]. Additionally, interconnectivity is required among everything that is connected in IoT that needs integration of the networking components [27]. This openness of sensors in IoT scenarios also comes with the different attacks and vulnerabilities and these security threats are identified by the authors in [28].

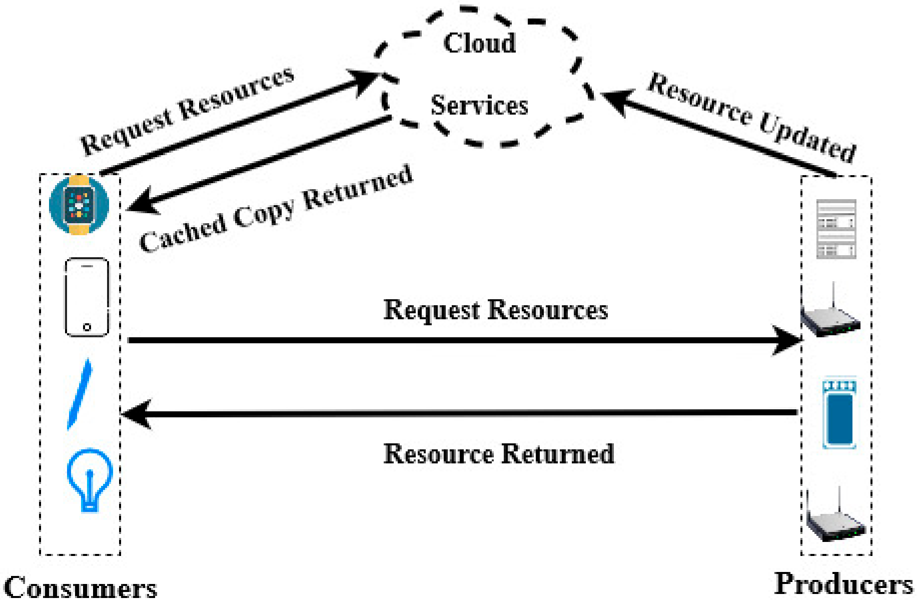

Figure 3. Object security framework for IoT (OSCAR) framework.

In [29], a study was made on considering the effective security solutions for IoT. The above solutions are independent of the platform and contribute to energy savings. IoT protection in smart healthcare applications, alongside energy efficiency, is a key issue. Password conceiving is the most common thing that can be done with a weak security framework in such applications. In [30], the user's privacy in applications with IoT is maintained through a certain password strengthening technique.

IoT devices are a vein of security for the operation of IoT to ensure the validity of devices and data gathered from these devices. In the literature, many authentication methods to verify the data collected from the devices are available to ensure the authenticity of the application. To improve the third party's faith in the IoT system, the data can be collected carefully. In [31], a scheme for sensing the sensed data is proposed that is based on policy and confidence. The suggested scheme, Real Alert, shows the trust of both the systems and the collected data. It increases the user's conviction.

In [32], the author introduced an IoT application validation protocol (AAoT). This scheme involves no changes to existing micro control units. The disadvantage of this system is that dynamic vulnerabilities cannot be corrected. The Authenticated Key Exchange system [33], which provides protection to side-channel attacks and is versatile in key certificate management, is still vulnerable to the leakage of random secret values.

New threats are being inserted at every point in existing IoT operations. Consequently, a SCADA model was introduced in [34] for threat detection. For detection of attacks, detection models using vector support machines and networks of deep faith are used.

To deal with the information for attack detection, the current attacks involve other intelligent methods and one of their intelligent security strategies is proposed [35]. In [36], the authors addressed background history, concerns, holes, and obstacles in IoT that contribute to these attacks. Devices are authenticated by Zigbee technology between devices in [37]. This is important because IoT devices are heterogeneous and therefore need a common platform for security arrangements between devices.

Therefore, based on related work, it can be concluded that the existing framework for IoT security implements solutions that are heavyweight in context to their key size, the complex structure of each round, and key schedules. The existing frameworks lack efficiency regarding security, throughput, and the delay in packet transmission.

3. Proposed Hybrid Logical Security Framework (HLSF)

The proposed HLSF contends to offer security services like confidentiality, integrity, and authentication. For this, HLSF is divided into three phases. The first phase is for the registration of the new devices that join the network. Second phase lightweight authentication is designed for providing a mechanism with which every device has to authenticate itself to the centralized server. Once the authentication is over, the last phase concentrates on the security of data in transit when different devices communicate with each other.

The entire HLSF is executed by taking an example of inventory automation, with a coordinating unit (CU), items, a database, an inventory server (IS), and an internet service provider as the components. The architecture representing the communication flow among these components is represented in Figure 4.

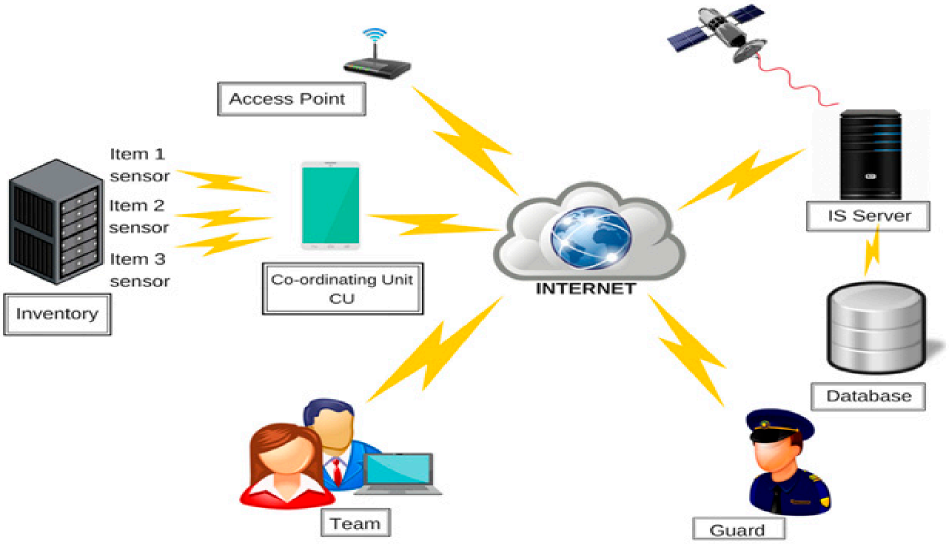

Figure 4. Hybrid logical security framework (HLSF) architecture.

3.1. Security Features

The proposed HLSF offers authentication, data confidentiality, and validation by using efficient mechanisms for each. The whole process is categorized into three phases, namely registration, then authentication, and finally data security. The notations used in every algorithm are described in Table 1.

Table 1. Notations used in Hybrid Logical Security Framework (HLSF).

Symbol	Description
C&N_C	Coordinating Unit (CU) and Number of CU
ID_C&ID_S	Identity of CU and Identity of Server
TID_C	Temporary Identity of CU
IS	Inventory Server
SN	Sequence Number
UID	Unique IDs
K_S	Key Shared between CU and Server
T_V	Temporary Variable
K_a	Alternate Keys

- **Registration Phase**

When a new device joins a network, the credentials are logged first to the server using the key-sharing system as shown in Figure 5.

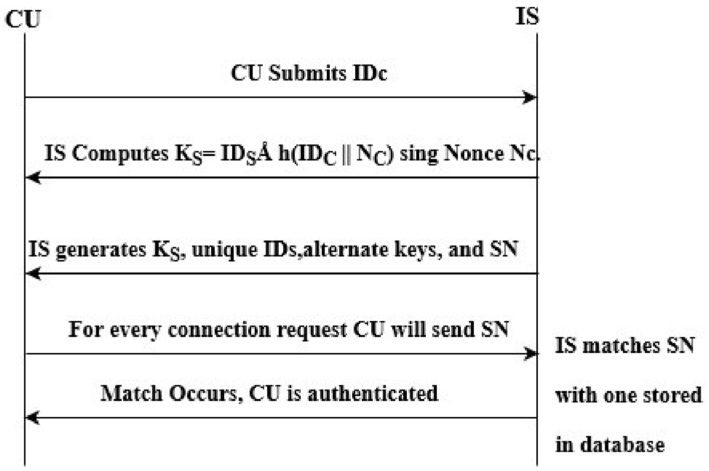
CU
IS
CU Submits IDc
IS Computes K_S= ID_SÅ h(ID_C || N_C) sing Nonce Nc
IS generates K_S, unique IDs,alternate keys, and SN
For every connection request CU will send SN
IS matches SN
with one stored
in database
Match Occurs, CU is authenticated

Figure 5. Registration process of HLSF.

In the registration phase, whenever a new CU joins, it has to submit its identity to IS. IS, further using its own identity, the identity of CU and a nonce value, computes a secret key K_S, along with unique IDs, alternate keys, and a unique sequence number for that particular CU. Every time a connection request is made by CU to IS, the sequence umber of CU is matched with the one stored in IS. If a match occurs, CU is authenticated at the server otherwise it makes use of alternate keys for proving its identity to IS.

- **Authentication Phase**

Once the device receives the credentials, the mutual authentication between the client and the server is completed, as shown in Figure 6 before any contact is initiated.

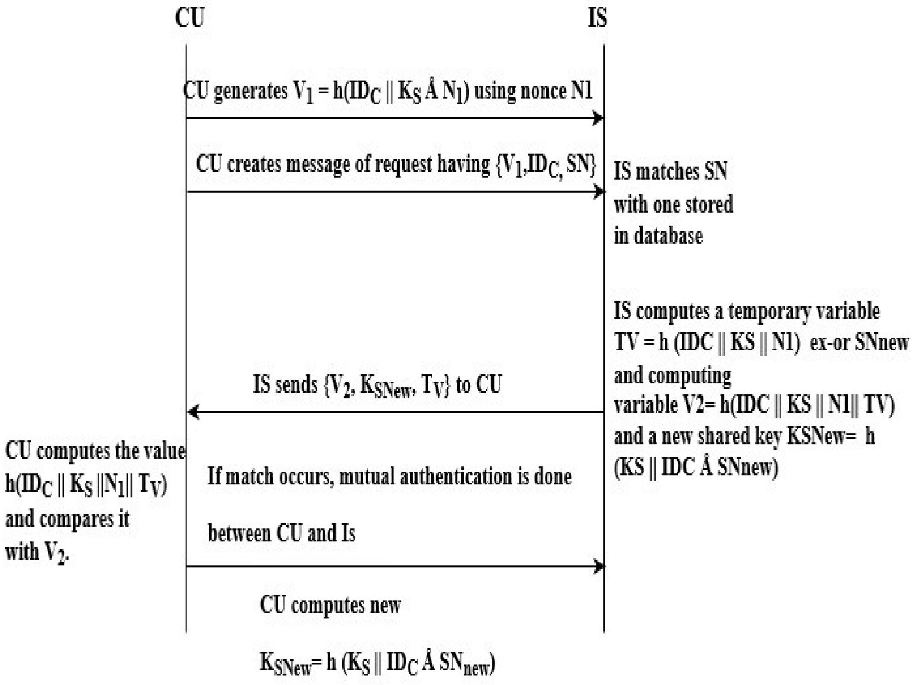
CU
IS
CU generates V_1 = h(ID_C || K_S Å N_1) using nonce N1
CU creates message of request having {V_1,ID_C, SN}
IS matches SN
with one stored
in database
IS computes a temporary variable
TV = h (IDC || KS || N1) ex-or SNnew
and computing
variable V2= h(IDC || KS || N1|| TV)
and a new shared key KSNew= h
(KS || IDC Å SNnew)
IS sends {V_2, K_{SNew}, T_V} to CU
CU computes the value
h(ID_C || K_S ||N_1|| T_V)
and compares it
with V_2.
If match occurs, mutual authentication is done
between CU and Is
CU computes new
K_{SNew}= h (K_S || ID_C Å SN_{new})

Figure 6. Authentication process of HLSF.

The mutual authentication process starts between CU and IS, where CU first generates a variable computing hash of its identity, a nonce, and an already shared secret key. CU then sends an authentication request consisting of a variable, its ID, and the SN it already has. IS, on receiving the request, first matches the SN received by CU; if a match occurs, IS initiates the process of authentication. For this, IS generates a temporary variable, again using a hash function on the variable received by CU, a secret key and a nonce. CU extracts its variable from a received message from IS. If a match occurs, IS is also authenticated at CU. Once this mutual authentication is done, further data communication starts.

- **Securing Data in HLSF**

Once the device is authenticated, the data communicated to and from the device are made secure using a security algorithm, as shown in Figure 7.

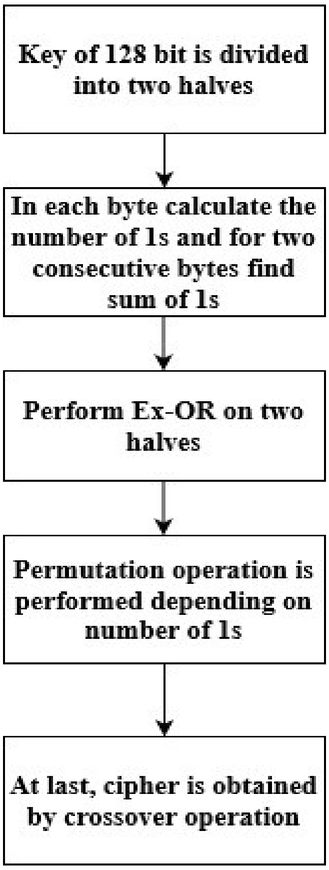
Key of 128 bit is divided into two halves
In each byte calculate the number of 1s and for two consecutive bytes find sum of 1s
Perform Ex-OR on two halves
Permutation operation is performed depending on number of 1s
At last, cipher is obtained by crossover operation

Figure 7. Data security process of HLSF.

Data security in communication is achieved by performing computation and permutation operations on data being shared. First, the key is divided into two halves to perform computations like finding 1s and the sum of 1s. Later, an Ex-OR operation is done on two separate halves. Finally, the ciphertext is fetched by permutating the halves and using crossover operation.

3.2. Security Analysis of HLSF

Proposed HLSF works in three different ways to overcome attacks; that is, during registration, authentication, and then finally during data transit. Each phase of HLSF tends to make the whole framework secure by preventing attacks based on its working mechanism. Security analysis of HLSF will explore the attacks that are restricted by each phase and hence makes HLSF less vulnerable to attacks.

- Attack Resistance during Registration

This phase of registration is used to generate a secret key that will be further used during the authentication and data security phases. Any CU who wishes to be part of the network has to register itself to the server. There is a sequence number assigned to CU by IS during registration. Whenever a communication initiates, the CU has to present that sequence number. If an intruder tries to connect to IS, the IS will ask for a sequence number or any alternate key which the intruder will not have. When there is a mismatch, IS will not allow an intruder to join the network. This concept of matching SN will prevent a denial of service attack against the server, as an intruder will not be able to connect to the server.

- Attack Resistance during Authentication

Mutual authentication is performed among CU and IS using multiple parameters, such as a secret key, a random variable, and an SN that is generated in the registration phase. In the authentication phase, the entire security process depends on multiple parameters for generating authentication messages at CU as well as for generating a response message from IS. Even if the secret key generated in the registration phase is compromised by an intruder, they will still not have the SN, and thus cannot authenticate themselves to the IS. This will again help in preventing denial of service attacks and is also non-vulnerable to the man in the middle attack.

- Attack Resistance during Data Transit

This phase is used to transmit data securely. For this, certain operations like EX-OR, permutation, and cross over operations are performed on the secret key and the plain text to figure out the ciphertext. EX-OR operation is used as it is reversible as well as the output calculated depends on both the halves. The simple but tricky mechanism of data security abstains from compromising attacks and replay attacks as each time a new secret key is created for information transmission.

4. Performance Comparison of COAP, OSCAR, and HLSF

This section evaluates and compares the performance of existing frameworks CoAP, OSCAR with the proposed framework HLSF. First, the security effectiveness of the frameworks is tested by finding the memory requirements, energy overhead, computational overhead, communicate rate, and denial of service attack. Later, the effectiveness of the overall framework from authentication, data collection, data security, data mining, and decision making is tested in terms of throughput, latency, and packet delivery ratio. Considering the heterogeneous nature of frameworks, certain research assumptions are made and realized for the performance evaluation.

4.1. Simulation Tool and Simulation Parameters

To evaluate the performance of HLSF, CoAP, and OSCAR, the COOJA simulator developed by Adam Dunkels in 2002 supported on CONTIKI OS is used. COOJA offers an environment where sensor motes can connect, communicate, and share data. While the data are shared among the motes, each security framework is implemented on data to measure its performance. The simulation parameters used to carry out this evaluation are represented in Table 2 below.

Table 2. Simulation parameters.

Parameter Name	Value
Radio medium	Unit Disk Graph Medium (UDGM)
Transmission range	50 m
Inference range	100 m
Type of Channel	Wireless
Nodes Position	Random
Sensing interval	10 s
MAC protocol	Contiki MAC
Routing protocol	RPL
Map Area	1000 × 1000 m^2

4.2. Memory Requirements

The random-access memory (RAM) and read-only memory (ROM) are evaluated for CoAP, OSCAR, and the proposed HLSF, using COOJA as the simulator. Figure 8 represents the memory requirements in percentage by considering the total available memory in CoAP, OSCAR, and HLSF.

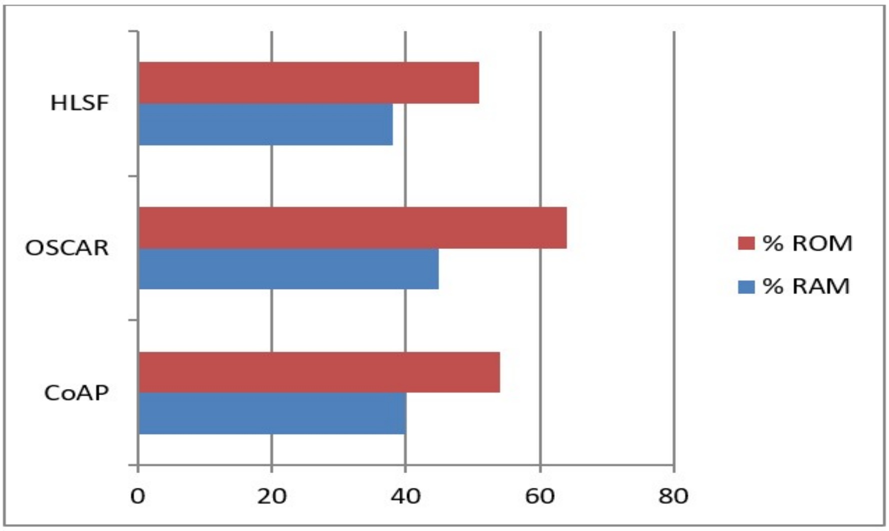

Figure 8. Percentage memory requirement for CoAP, OSCAR, and HLSF.

From Figure 8, it can be concluded that the ROM requirement of HLSF is 3 percent less than CoAP and 13 percent less than OSCAR. On the other hand, the RAM requirement of HLSF is 2 percent less than CoAP and 7 percent less than OSCAR. Therefore, it can be deduced that the overall memory requirements of HLSF are less as compared to those of OSCAR and CoAP.

4.3. Energy Overhead

The energy overhead of the security framework has a direct impact on the lifetime of the sensors and will ultimately impact the transmission rate of the application. Therefore, if the energy overhead increases for a particular application, its lifetime gradually decreases. A more complex security framework possesses more energy overhead. Figure 9 represents the energy overhead calculated in millijoules (MJ) for CoAP, OSCAR, and HLSF.

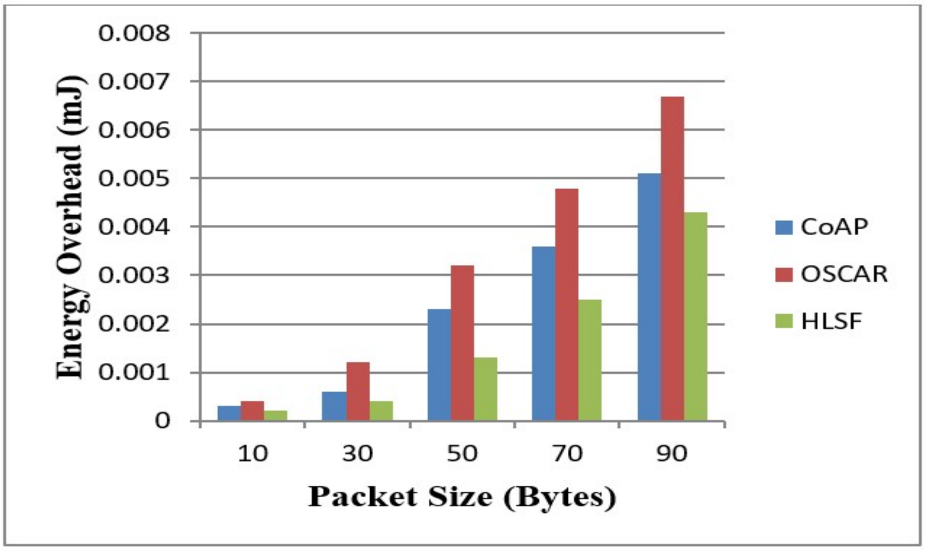

Figure 9. Energy overhead for CoAP, OSCAR, and HLSF.

Figure 9 reflects that energy overhead depends on the packet size. With an increase in packet size, energy overhead goes on increasing. Even at the largest packet size, the energy overhead of the proposed HLSF is 18 percent less than CoAP and 55 percent less than OSCAR.

4.4. Computational Overhead

Computational overhead is the excess time required by a security framework for offering security used in IoT applications. Figure 10 represents the computational overhead evaluated in milliseconds (ms) for CoAP, OSCAR, and HLSF.

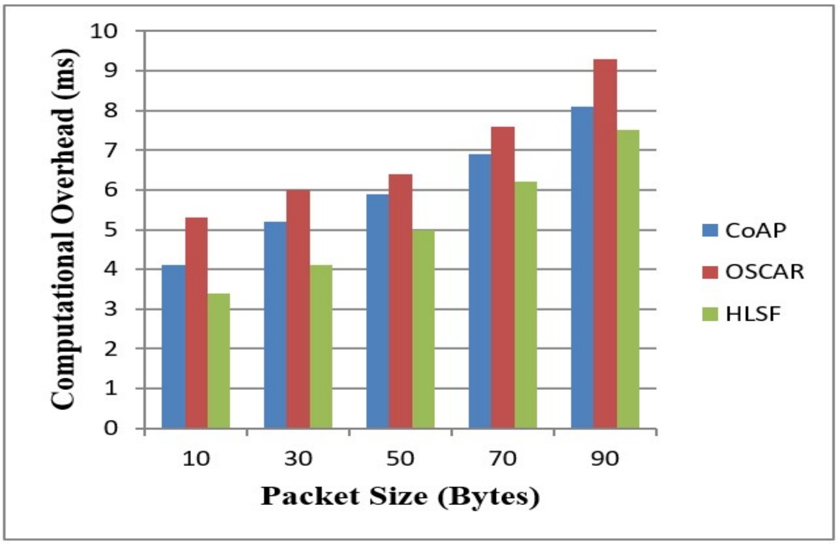

Figure 10. Computational overhead for CoAP, OSCAR, and HLSF.

Figure 10 reflects that the computational overhead depends on the packet size. With an increase in packet size, computational overhead goes on increasing. Even at the largest packet size, the computational overhead of proposed HLSF is 8 percent less than CoAP and 24 percent less than OSCAR.

4.5. Communication Rate

The communications rate is the number of packets sent over in one second. The communication rate is impacted by the type of security framework used for an application scenario. An application that does not offer any security will tend to have a greater communication rate than a security-oriented application. Figure 11 represents the communication rate of CoAP, OSCAR, and HLSF, considering the packet size as 64 bytes.

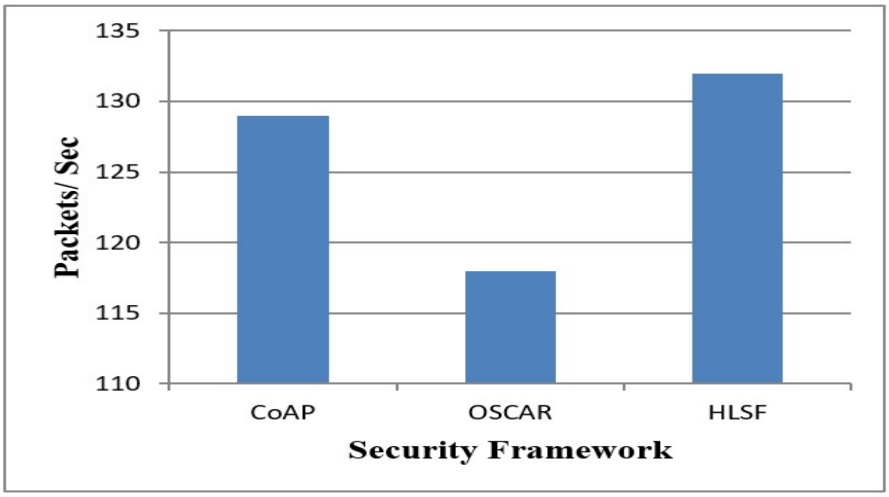

Figure 11. Communication Rate for CoAP, OSCAR, and HLSF.

From Figure 11, it can be concluded that the communication rate—that is, the number of packets sent per unit time—is more in HLSF, as it is more than 2 percent greater than that of CoAP and more than 10 percent greater than that of OSCAR.

The overall result is summarized below:

- With a fewer number of rounds required, the memory requirements of HLSF are less as compared to CoAP and OSCAR.
- HLSF has a less complex structure, as it does not work on an asymmetric algorithm, hence making its energy overhead less.
- HLSF takes less time to encrypt the data as the round keys are generated at each round and are of optimal size, making its computational overhead smaller.
- Even if the number of rounds is less and the structure is less complex and but still tricky, each round generates a unique set of keys. Thus, this makes it less vulnerable to denial of service attack, as it becomes difficult for the intruder to get over the frequently changing key. Moreover, every time intruder has to prove its authenticity to the server.
- HLSF with a security mechanism offered still has a high communication rate, as fewer packets are lost or attacked in transmission.

5. Conclusions

With the proliferation of advanced technology, the environment is facing more challenges due to e-waste and risky emissions. It becomes very essential that more research must take into consideration green IoT to preserve our environment and to make society more astute and greener. As the essential factors of ICT become more advanced, the things around us will get smarter, performing explicit endeavors in a self-administering way, rendering the new sort of green communication among human and things and between entities themselves. Providing security and network functionalities with a preferable decreased energy usage is one of the requirements of next generation IoT. In light of this, the hybrid logical security framework (HLSF) has been proposed, consisting of three phases:

namely, registration, authentication, and data security. HLSF provides a lightweight security solution that requires a lower key size and frequently changes the pattern of the key. We have simulated and compared our proposed method with two well-known frameworks, namely CoAP and OSCAR. The simulation result shows that the memory requirement for HLSF is 3% and 13% less as compared to CoAP and OSCAR, respectively. HLSF outperforms the CoAP and OSCAR in terms of computational and energy overhead. The computational overhead of the proposed method is 8% and 24% less in comparison to CoAP and OSCAR, respectively. The energy requirement of HLSF is 18% less than CoAP and 55% less than OSCAR. Moreover, the throughput of HLSF is more than 2% greater than that of CoAP and 10% more than that of OSCAR. Thusly, it is evident that HLSF provides better network functionalities with a low overhead as compared to CoAP and OSCAR. In order to make the system more efficient and smarter, further research can be carried out by investigating the use of data mining and enabling smart decision making for innovative dynamics in IoT.

Author Contributions: Formal analysis, A.M. and U.G.; Investigation, G.N.N.; Resources, A.S.M.S.H. and V.M.; Validation, J.J.P.C.R.; Writing—original draft, S.V. and I.B.; Writing—review & editing, K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by research funds of Jeonbuk National University in 2020. This work is partially funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/EEA/50008/2020; and by Brazilian National Council for Scientific and Technological Development (CNPq) under Grant No. 309335/2017-5.

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

References

1. Verikoukis, C.; Minerva, R.; Guizani, M.; Datta, S.K.; Chen, Y.; Muller, H.A. Internet of Things: Part 2. *IEEE Commun. Mag.* **2017**, *55*, 114–115. [CrossRef]
2. Silva, J.S.; Zhang, P.; Pering, T.; Boavida, F.; Hara, T.; Liebau, N.C. People-Centric Internet of Things. *IEEE Commun. Mag.* **2017**, *55*, 18–19. [CrossRef]
3. Hasan, M.; Islam, M.M.; Islam, I.; Hashem, M.M.A. Attack and Anomaly Detection in IoT Sensors in IoT Sites Using Machine Learning Approaches. *Internet Things* **2019**, *7*, 100059. [CrossRef]
4. Yang, Y.; Wu, L.; Li, G.Y.L.; Zhao, H. A survey on security and privacy issues in internet-of-things. *IEEE Internet Things J.* **2017**, *4*, 1250–1258. [CrossRef]
5. Ling, Z.; Luo, J.; Xu, Y.; Gao, C.; Wu, K.; Fu, X. Security vulnerabilities of the internet of things: A case study of the smart plug system. *IEEE Internet Things J.* **2017**, *4*, 1899–1909. [CrossRef]
6. Cheng, C.; Lu, R.; Petzoldt, A.; Takagi, T. Securing the Internet of Things in a quantum world. *IEEE Commun. Mag.* **2017**, *55*, 116–120. [CrossRef]
7. Allhoff, F.; Henschke, A. The Internet of Things: Foundational ethical issues. *Internet Things* **2018**, *1*, 55–66. [CrossRef]
8. Kawamoto, Y.; Nishiyama, H.; Kato, N.; Shimizu, Y.; Takahara, A.; Jiang, T. Effectively collecting data for the location-based authentication in the Internet of Things. *IEEE Syst. J.* **2017**, *11*, 1403–1411. [CrossRef]
9. Garcia-de-Prado, A.; Ortiz, G.; Boubeta-Puig, J. COLLECT: Collaborative Context-aware service-oriented architecture for intelligent decision-making in the Internet of Things. *Expert Syst. Appl.* **2017**, *85*, 231–248. [CrossRef]
10. Fussler, C.; James, P. *Eco-Innovation: A Break thorough Discipline for Innovation and Sustainability*; Pitman: London, UK, 1996.
11. Correia, E.; Carvalho, H.; Azevedo, S.G.; Govindan, K. Maturity models in supply chain sustainability: A systematic literature review. *Sustainability* **2017**, *9*, 64. [CrossRef]
12. Li, W.; Xu, J.; Zheng, M. Green governance: New perspective from open innovation. *Sustainability* **2018**, *10*, 3845. [CrossRef]
13. Wang, J.; Gao, Y.; Zhou, C.; Sherratt, R.S.; Wang, L. Optimal Coverage Multi-Path Scheduling Scheme with Multiple Mobile Sinks for WSNs. *Comput. Mater. Contin.* **2020**, *62*, 695–711. [CrossRef]
14. Wang, J.; Gao, Y.; Yin, X.; Li, F.; Kim, H. An Enhanced PEGASIS Algorithm with Mobile Sink Support for Wireless Sensor Networks. *Topol. Control Emerg. Mobile Netw.* **2018**. [CrossRef]

15. Min, Z.; Yang, G.; Wang, J.; Kim, G. A Privacy-preserving BGN-type Parallel Homomorphic Encryption Algorithm Based on LWE. *J. Internet Technol.* **2019**, *20*, 2189–2200.
16. Choo, K.R.; Gritzalis, S.; Park, J.H. Cryptographic Solutions for Industrial Internet-of-Things: Research Challenges and Opportunities. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3567–3569. [CrossRef]
17. Eclipse Organization. Mqtt and Coap, IoT Protocols. Available online: http://www.eclipse.org/community/eclipse_newsletter/2014/february/article2.php (accessed on 5 February 2014).
18. Shelby, Z.; Hartke, K.; Bormann, C. *The Constrained Application Protocol (CoAP)*; Standards Track RFC 7252; Center for Computing Technologies (TZI), University of Bremen: Bremen, Germany, June 2014.
19. Mišić, J.; Ali, M.Z.; Mišić, V.B. Architecture for IoT Domain with CoAP Observe Feature. *IEEE Internet Things J.* **2018**, *5*, 1196–1205. [CrossRef]
20. Mišić, J.; Mišić, V.B. Proxy cache maintenance using multicasting in CoAP IoT domains. *IEEE Internet Things J.* **2018**, *5*, 1967–1976. [CrossRef]
21. Correia, N.; Sacramento, D.; Schütz, G. Dynamic aggregation and scheduling in CoAP/observe-based wireless sensor networks. *IEEE Internet Things J.* **2016**, *3*, 923–936. [CrossRef]
22. Betzler, A.; Gomez, C.; Demirkol, I.; Paradells, J. CoAP congestion control for the internet of things. *IEEE Commun. Mag.* **2016**, *54*, 154–160. [CrossRef]
23. Son, S.; Kim, N.; Lee, B.; Cho, C.; Chong, J. A time synchronization technique for coap-based home automation systems. *IEEE Trans. Consum. Electron.* **2016**, *62*, 10–16. [CrossRef]
24. Park, C.; Park, W. A Group-Oriented DTLS Handshake for Secure IoT Applications. *IEEE Trans. Autom. Sci. Eng.* **2018**, *99*, 1–10. [CrossRef]
25. Vucinic, M.; Tourancheau, B.; Rousseau, F.; Duda, A.; Damon, L. OSCAR: Object Security Architecture for the Internet of Things. In Proceedings of the 2014 IEEE 15th International Symposium, Sydney, Australia, 19 June 2014; pp. 3–16.
26. Aly, M.; Khomh, F.; Haoues, M.; Quintero, A.; Yacout, S. Enforcing Security in Internet of Things Frameworks: A Systematic Literature Review. *Internet Things* **2019**, *6*, 100050. [CrossRef]
27. Younis, M. Internet of everything and everybody: Architecture and service virtualization. *Comput. Commun.* **2018**, *131*, 66–72. [CrossRef]
28. Alaba, F.A.; Othman, M.; Hashem, I.A.T.; Alotaibi, F. Internet of Things security: A survey. *J. Netw. Comput. Appl.* **2017**, *88*, 10–28. [CrossRef]
29. Hellaoui, H.; Koudil, M.; Bouabdallah, A. Energy-efficient mechanisms in the security of the internet of things: A survey. *Comput. Netw.* **2017**, *127*, 173–189. [CrossRef]
30. He, D.; Ye, R.; Chan, S.; Guizani, M.; Xu, Y. Privacy in the Internet of Things for Smart Healthcare. *IEEE Commun. Mag.* **2018**, *56*, 38–44. [CrossRef]
31. Li, W.; Song, H.; Zeng, F. Policy-based secure and trustworthy sensing for the internet of things in smart cities. *IEEE Internet Things J.* **2018**, *5*, 716–723. [CrossRef]
32. Feng, W.; Qin, Y.; Zhao, S.; Feng, D. AAoT: Lightweight attestation and authentication of low resource things in IoT and CPS. *Comput. Netw.* **2018**, *134*, 167–182. [CrossRef]
33. Ruan, O.; Zhang, Y.; Zhang, M.; Zhou, J.; Harn, L. After-the-fact leakage-resilient identity based authenticated key exchange. *IEEE Syst. J.* **2018**, *12*, 2017–2026. [CrossRef]
34. Huda, S.; Yearwood, J.; Hassan, M.M.; Almogren, A. Securing the operations in SCADA-IoT platform-based industrial control system using ensemble of deep belief networks. *Appl. Soft Comput.* **2018**, *71*, 66–77. [CrossRef]
35. Miloslavskaya, N.; Tolstoy, A. Internet of Things: Information security challenges and solutions. *Cluster Comput.* **2019**, *22*, 103–119. [CrossRef]
36. Adat, V.; Gupta, B.B. Security in Internet of Things: Issues, challenges, taxonomy, and architecture. *Telecommun. Syst.* **2018**, *67*, 423–441. [CrossRef]
37. Alshahrani, M.; Traore, I.; Woungang, I. Anonymous mutual IoT inter-device authentication and key agreement scheme based on the ZigBee technique. *Internet Things* **2019**, *7*, 100061. [CrossRef]