*future internet*

MDPI

*Article*

# Securing SDN-Based IoT Group Communication

**Bander Alzahrani [1] and Nikos Fotiou [2,*]**

[1] Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia; baalzahrani@kau.edu.sa

[2] Mobile Multimedia Laboratory, Department of Informatics, School of Information Sciences and Technology, Athens University of Economics and Business, Patision 76, 10434 Athens, Greece

* Correspondence: fotiou@aueb.gr

**Abstract:** IoT group communication allows users to control multiple IoT devices simultaneously. A convenient method for implementing this communication paradigm is by leveraging software-defined networking (SDN) and allowing IoT endpoints to "advertise" the resources that can be accessed through group communication. In this paper, we propose a solution for securing this process by preventing IoT endpoints from advertising "fake" resources. We consider group communication using the constrained application protocol (CoAP), and we leverage Web of Things (WoT) Thing Description (TD) to enable resources' advertisement. In order to achieve our goal, we are using linked-data proofs. Additionally, we evaluate the application of zero-knowledge proofs (ZKPs) for hiding certain properties of a WoT-TD file.

**Keywords:** crowd management; software-defined networking; linked-data signatures; Web of Things; zero-knowledge proofs



**Citation:** Alzahrani, B.; Fotiou, N. Securing SDN-Based IoT Group Communication. *Future Internet* **2021**, *13*, 207. https://doi.org/10.3390/fi13080207

Academic Editor: Christoph Stach

Received: 9 July 2021
Accepted: 3 August 2021
Published: 9 August 2021



## 1. Introduction

The Internet of Things (IoT) considers "unconventional" communication paradigms such as "publish–subscribe" or "one-to-many" communication; in this paper, we focus on the latter paradigm, which is usually referred to as group communication. Although this paradigm is not typical in mainstream communication systems, we postulate that this is not the case for the IoT. We consider as a use case the crowd monitoring system presented in [1]. This system includes gas and ultrasonic sensors, UAVs equipped with cameras and LiDARs, as well as CCTV systems (see also Figure 1). In this system, tasks such as "collect all measurements in area X" or "turn on all cameras in area Z" are not uncommon scenarios, and the reasonable approach for implementing them is using group communication.

Group communication using the constrained application protocol (CoAP) [2] is a promising direction, which is impeded by the lack of adoption of IP Multicast, however. On the other hand, interconnecting IoT devices over software-defined networking (SDN)—such as in the architecture presented in [1]—enables alternative approaches for implementing group communication that removes the need for IP multicast and enables "self-organizing" IoT groups, where IoT endpoints can "advertise" the CoAP URIs of their resources, and groups can be automatically created based on these advertisements. Therefore, it is obvious that this advertisement process must be protected; otherwise, malicious entities may "pollute" the network with "fake" advertisements, affecting this way the group formation process.

In this paper, we provide a solution to this problem by allowing only authorized endpoints to perform advertisements. From a high-level perspective, we consider that each endpoint "represents" its available resource using a JSON-encoded file and by following the W3C Web of Things, Thing Description (WoT-TD) specifications [3]. This WoT-TD file is signed by a trusted service provider, and it is included in the advertisements, together with proof of ownership. The recipients of such an advertisement can then easily verify its validity. In this paper, we make the following contributions:

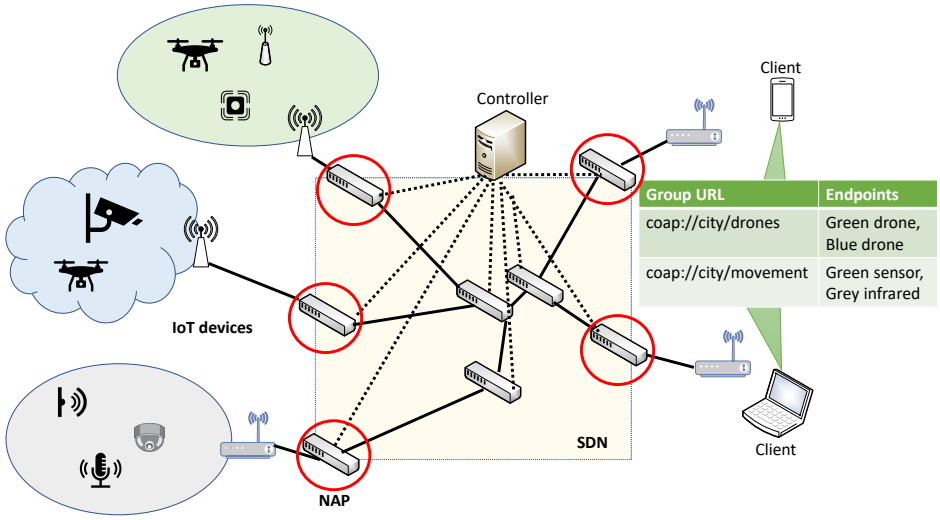


**Figure 1.** An overview of the entities of the proposed solution.

- We design an IoT onboarding process that ensures that only authorized IoT devices participate in a group;
- We leverage linked-data signatures to provide advertisement validity and proof of ownership;
- We extend our solution to support selective advertisement of resources using zero-knowledge proofs.

The remainder of this paper is organized as follows: In Section 2, we introduce the technologies used as building blocks of our system. In Section 3, we detail the design of our solution, and in Section 4, we present its implementation and evaluation. We discuss related work in this area in Section 5, and we conclude our paper in Section 6.

## 2. Background

### *2.1. SDN and Bloom Filter-Based Forwarding*

Software-defined networking (SDN) [4] is a technology that allows a centralized entity, known as the "network controller" (or simply controller) to control programmable switches. SDN switches forward packets based on rules defined by the controller. In particular, in order for a switch to determine how to handle an incoming packet, it either queries the controller or uses a set of rules stored in the switch using a protocol such as OpenFlow [5].

SDN can be used for implementing Bloom filter [6]-based packet forwarding [7]. This type of forwarding enables multicast communication in an efficient, fast, and stateless way. From a high-level perspective, the solution in [7] assumes that each outgoing interface of an SDN switch is identified by a bitstring identifier; then, it uses a Bloom filter to encode the identifiers of all interfaces through which a packet should be forwarded; finally, it stores this forwarding identifier in the IPv6 address field of the packet. SDN switches are preconfigured with rules that allow them to decide the outgoing interface of each incoming packet simply by "ORing" the packet's forwarding identifier with the identifiers of all outgoing interfaces.

### *2.2. CoAP and CoAP Group Communication*

CoAP [8] is a lightweight protocol, designed to be the "HTTP of the IoT." CoAP resources are identified by a URI scheme, similar to HTTP URIs, and the CoAP interaction model is similar to the client–server model of HTTP. Therefore, IoT endpoints act as CoAP "servers", exposing one or more CoAP URIs that can be accessed by CoAP "clients" using a suitable CoAP "method" .

CoAP group communication is a CoAP extension that allows CoAP clients to retrieve (or set) resources from a group of CoAP servers, e.g., retrieve the temperature measurements from all sensors of a building, turn on and off all the lights of a smart city, etc. An

approach for realizing CoAP group communication is using IP multicast (Section 2 of [2]). With this approach, CoAP servers belonging to the same group join an IP multicast address, and CoAP clients learn the IP multicast address of a group using DNS resolution. Then, CoAP clients can send CoAP requests to an IP multicast address and receive the corresponding response(s) using unicast. Nevertheless IP multicast is not the only option; other underlay networking architectures can be used instead. For example, as as we discuss in the following section, our solution relies on SDN to implement one-to-many communication.

### 2.3. Web of Things

The goal of W3C's Web of Things (WoT) working group is to improve the interoperability and usability of the Internet of Things (IoT) [9] by specifying universal means for accessing IoT devices. This goal is achieved by providing building blocks that leverage and extend existing, standardized Web technologies in the context of IoT. Such a building block is the Thing Description specification draft [3].

A Thing Description (WoT-TD) is a JSON-LD [10] document that describes the "metadata" and "interfaces" of a "thing", where a thing can be a physical IoT device or a virtual entity that is composed of multiple IoT devices. An interface can be a "property", an "action", or an "event", that can be accessed using a Web technology such as CoAP or HTTP. A WoT-TD describes how these interfaces can be accessed by specifying suitable URIs, security policies, and other information that can be used by an interested client.

Being encoded using JSON-LD, a WoT-TD includes a `context` property, which is an array of URLs pointing to documents that include "vocabulary" terms. All WoT-TD include the "https://www.w3.org/2019/wot/td/v1" context, but additional contexts can be added, allowing the extension of the WoT-TD vocabulary (see also Section 7.1 of [3] for more information).

## 3. Overview

### 3.1. System Entities and Security Assumptions

Our solution considers an SDN network that interconnects IoT *endpoints* acting as CoAP servers with IoT service clients. Each IoT endpoint owns an Ed22519 public key [11], denoted by $Endpoint_{ID}$. Network operators maintain a list of $Endpoint_{ID}$ identifiers belonging to authorized IoT devices. IoT devices are connected to the SDN network through an access device; the type of this device depends on the available communication technology (e.g., WiFi, ZigBee, LoRa, etc.); nevertheless, our solution is oblivious to used technology. Access devices are connected to edge switches acting as the network attachment point (NAP), and it is assumed that there are mechanisms that allow NAPs to access this list of authorized IoT devices. Therefore, it should be not possible for an attacker to join a network by impersonating an authorized IoT device inside the SDN network. Similarly, the network operator owns a well-known Ed22519 public key, denoted by $Network_{ID}$, which acts as the root of trust in our system; all endpoints are preconfigured with this key, and all NAPs can generate signatures using the private key that corresponds to a $Network_{ID}$. Whenever a key is included in a text-based file, we are using its Base64url encoding [12].

IoT devices provide access to resources. We focus on resources that can be accessed using CoAP and CoAP group communication.

Additionally, we consider that the SDN controller knows the full network topology, and it is capable of constructing forwarding paths from an $Endpoint_{ID}$ toward one or more $Endpoint_{ID}$ identifiers (see also Section 2.1). These paths are identified by a Bloom filter-based identifier denoted as $Fwd_{ID}$. Finally, we consider that there is a well-known $Fwd_{ID}$ that can be used by endpoints to broadcast packets in the network.

Our solution is focused on IoT endpoints acting as CoAP servers; for this reason, we neither consider clients as part of our threat model nor are we concerned with client-facing security operations such as client authentication and authorization.

### *3.2. IoT Device Onboarding*

Each IoT device is preconfigured with a WoT-TD file for the resources it provides. An example of a WoT-TD is one that includes an ultrasonic sensor deployed in a stadium, which can be seen in Listing 1. Lines 1–5 define the *context* of the WoT-TD file and include the identifier of the IoT device, i.e., the $Endpoint_{ID}$. This example also includes an *action* used for "turning off" the sensor (line 8). This action can be invoked using either plain CoAP (lines 10–13) or CoAP group communication (lines 14–23). As it can be observed, this action can be invoked through two different groups, one representing "all sensors of the stadium" (line 15) and another representing "all sensors of a city" (line 20).

**Listing 1.** An example of a WoT-TD file.

```
1   {
2   "context": "https://www.w3.org/.../v1",
3   ...
4   "id": Endpoint_ID,
5   }
6   "properties": {...},
7   "actions": {
8     "turnoff_sensor": {
9       "forms":
10        [{
11          "href":"coap://gate7.stadium/sensor1/turnoff",
12          "cov:methodName": "POST''
13        },
14        {
15          "href'':"coap://stadium/sensors/turnoff",
16          "cov:methodName'': "POST''
17          "subprotocol":"cov:group"
18        },
19        {
20          "href'':"coap://city1/sensors/turnoff",
21          "cov:methodName'': "POST''
22          "subprotocol":"cov:group"
23        }]
24    }
25  },
26  "events": {...}
27  }
```

In order for an IoT device to join the network, it establishes a (D)TLS communication channel with a NAP. The (D)TLS handshake uses the "client authentication" option. The goal of this handshake is to allow the IoT device to verify that the NAP knows the private key that corresponds to $Network_{ID}$, and the NAP to verify that the IoT device is the owner of $Endpoint_{ID}$.

As a next step, the IoT device sends its WoT-TD file to the NAP, and the NAP verifies that it includes the same $Endpoint_{ID}$ used during the handshake, as well as that the $Endpoint_{ID}$ is in the list of authorized identifiers. Then, the NAP signs the WoT-TD file using a *linked-data proof* (LDP) [13]. An LDP is a mechanism for ensuring the authenticity and integrity of linked-data documents, such as WoT-TD files, which is extensible and supports contemporary cryptographic solutions, such as zero-knowledge proofs (ZKPs). An LDP is encoded using JSON, and an example of an LDP used in our system is included in Listing 2. As it can be seen, line 2 defines the type of the proof, which, in our example, is an EdDSA signature [11]; line 3 includes a timestamp indicating the proof creation time; line 4 includes information that can be used for verifying the proof, which, in our case, is the $Network_{ID}$; line 5 includes the

purpose of the proof, which, in this listing, is to provide an "assertion" about the integrity of the WoT-TD file; finally, line 6 is the actual proof value.

**Listing 2.** A linked-data proof used in our system.

```
1 {
2     "type": "Ed25519Signature2020",
3     "created": "2021−17−06T11:01:24Z",
4     "verificationMethod": Network_ID,
5     "proofPurpose": "assertionMethod",
6     "proofValue": "VqpLMweBrSxMY2x...aqA3Q1geV6"
7   }
```

The received proof is appended to the WoT-TD file. From this point on, the IoT device can participate in the rest of the operations of our system.

### *3.3. SDN-Based IoT Group Communication*

Our system adapts the solution presented in [14] for providing SDN-based IoT group communication and implements IoT group communication as a two-step process. The first step involves the *advertisement* of the available resources, and the second step implements the actual CoAP *group requests*.

#### 3.3.1. Resource Advertisement

IoT devices should advertise their resources by broadcasting their WoT-TD files. As a reminder, we assume a well-known $Fwd_{ID}$ that can be used for broadcasting. In order to protect advertisements from replay attacks, IoT devices generate a new LDP, similar to the one they have received by the NAP, which, however, includes a *nonce*, and is generated using the private key that corresponds to $Endpoint_{ID}$. Nonces in our system must not be reused within a specific time frame. This can be easily implemented by maintaining a list of used nonces: each nonce should remain in the list for the duration of the selected time frame, and devices must make sure that a nonce they send or receive is not included in that list.

Upon receiving an advertisement, clients validate the integrated LDPs. In particular, they validate that the advertisement includes an LDP that can be verified using $Network_{ID}$, which is "well known", and another that can be verified using $Endpoint_{ID}$ included in the WoT-TD file. Additionally, they verify that the latter proof is adequately fresh, and it includes a unique nonce. If all verifications are successful, each client updates a *lookup table* that includes mappings from CoAP group URIs to the corresponding $Endpoint_{ID}$ identifiers.

#### 3.3.2. CoAP Group Request

A CoAP client wishing to send a request to a CoAP group implements the related protocol described in [14]. From a high-level perspective the client executes the following steps:

1. From the lookup table, it retrieves the $Endpoint_{ID}$ identifiers of the CoAP servers that are members of the group;
2. If it knows a $Fwd_{ID}$ for all retrieved $Endpoint_{ID}$ identifiers, it proceeds to step 4;
3. It constructs a message that includes all $Endpoint_{ID}$ identifiers for which it does not know a $Fwd_{ID}$ and sends it to a "special" MAC address used for forcing SDN switches to forward a packet to the controller. The controller responds with a list of $Fwd_{ID}$ that is eventually returned back to the client;
4. It creates a new $Fwd_{ID}$ by ORing the $Fwd_{ID}$ identifiers of the retrieved $Endpoint_{ID}$ identifiers and forwards the CoAP request using the created $Fwd_{ID}$. Due to the properties of Bloom filter-based forwarding (see [7] for more details), the CoAP request will be forwarded to the appropriate IoT devices.

## 4. Implementation and Evaluation

### 4.1. Implementation

We implemented our solution using Eclipse's Thingweb node-wot (https://github.com/eclipse/thingweb.node-wot accessed on 5 August 2021) as an IoT endpoint, and libcoap library (https://libcoap.net/ accessed on 5 August 2021) for emulating CoAP clients. For the SDN underlay, we relied on the tools presented in [15], i.e., we used the Open vSwitch [16] SDN switch, the POX [17] SDN controller, and we emulated the network using the mininet network emulator [18]. Finally, we used the JSON-LD library (https://github.com/digitalbazaar/jsonld-signatures accessed on 5 August 2021) to generate and verify LDPs.

In order to not modify libcoap to support the used SDN-based group communication, we developed a CoAP proxy that implements the related protocols; CoAP clients wishing to send a request to a group simply forward their request to the proxy using plain CoAP (see Section 5.7 of [8]). Using this approach, group communication is implemented transparently from the used CoAP library. This is a useful property since it allows the use of our solution even with constrained IoT devices, acting as a CoAP client, although using CoAP libraries with limited functionality.

We measured the time required to generate and verify an LDP in a desktop PC equipped with an Intel-i5 CPU and 4GB RAM, running Xubuntu, and a Raspberry Pi 2 Model B Rev 1.1 with a 900 MHz quad-core ARM Cortex-A7 CPU and 1GB RAM, running Raspberry Pi OS. Table 1 shows the results. The size of the corresponding base64-encoded LDP is 508 bytes.

**Table 1.** LDP generation and verification times.

| Operation | Desktop | Raspberry Pi |
|---|---|---|
| LDP generation | 0.93 ms | 5.2 ms |
| LDP verification | 0.83 ms | 6.0 ms |

### 4.2. Security Evaluation

The security goal of our solution is to prevent "fake" advertisements. Indeed, with our solution, only authorized IoT endpoints can advertise WoT-TD files. Furthermore, because these WoT-TD files are singed, neither an active attacker nor the IoT endpoint itself can modify them.

An active attacker in our system is able to replay valid advertisements. Although, in general, replay attacks are prevented by the use of the *nonce*, there can be cases in which the replayed advertisement is received before the real one. In these cases, an endpoint will believe that the attacker is a legitimate IoT device and that the real advertisement was a replayed one. Although this attack cannot be prevented in a straightforward way, we argue that its impact is limited, and it can be easily detected and i mitigated. Advertisements in our system do not contain any location-specific information, since $Endpoint_{ID}$ identifiers are just public keys. Therefore, if an $Endpoint_{ID}$ still provides the advertised resources, the attack will have no impact apart from the added network overhead. Moreover, advertisements in our system are broadcasted; hence, it will be trivial for a monitoring entity to detect the replay attack. Finally, we consider an SDN-based architecture, in which a controller can remove any endpoint from the network in a straightforward manner.

Similarly, an attacker that has access to the private key that corresponds to an $Endpoint_{ID}$ can only sent *valid* advertisements of WoT-TD files belonging to the corresponding IoT device, i.e., since the $Endpoint_{ID}$ is included in the WoT-TD, the attacker cannot use the breached key to sign an advertisement of another IoT device. Therefore, the impact of this attack is similar to the impact of the replay attack.

From a security perspective, the most critical component of our solution is the private key that corresponds to $Network_{ID}$. If this key is compromised, then it must be revoked; hence, all endpoints must be reconfigured with the new $Network_{ID}$, and all LDPs must be regenerated.

### 4.3. Private Advertisements Using ZKPs

*Zero-knowledge proofs* (ZKP) are a class of proofs in which a *prover* can prove to a *verifier* the knowledge of a value without revealing what the value is [19,20]. In the context of our system, ZKPs can be used by a user in order to generate a WoT-TD that reveals only affordances that are accessed through CoAP group communication. In order to achieve this, the proof of the TD signature must have been generated using an appropriate signature algorithm. In our implementation, we used the BBS+ linked signature algorithm for this purpose [21]. This signature scheme makes use of BLS12-381 pairing-friendly keys [22]. In a WoT-TD that contains a BBS+ signature, a subset of the affordances can be hidden by "framing" the original WoT-TD in a JSON-LD frame [23]. A *JSON-LD frame* can be seen as a filter that, when applied to a JSON-LD document (e.g., a W3C-compliant VC), it outputs a new JSON-LD document that contains only a subset of the fields of the original document. Table 2 shows the time required to generate and verify an LDP proof, using the same endpoints as in Section 4.1, for a WoT-TD file that includes four affordances, three of which are hidden. For this purpose, we used node.js and the jsonld-bbs library, (https://github.com/mattrglobal/jsonld-signatures-bbs accessed on 5 August 2021), which handles JSON-LD objects and uses BLS12-381 keys to generate BBS+ ZKPs. The size of the corresponding base64-encoded LDP is 891 bytes.

**Table 2.** LDP generation and verification times when BBS+ is used.

| Operation | Desktop | Raspberry Pi |
|---|---|---|
| LDP | 174.6 ms | 999.2 ms |
| LDP verification | 74.3 ms | 1004.0 ms |

As can be seen from this Table, LDP generation and verification in the Raspberry Pi requires approximately 1 s. Nevertheless, we are using an old device and an implementation written in node.js; therefore, there is significant space for improvement. Additionally, these signatures have to be calculated every time a new WoT description file is advertised; this process does not have to take place often—its frequency can be in the order of hours.

## 5. Related Research

Many research efforts provide solutions for protecting the confidentiality of IoT group communication messages, e.g., using group object security for constrained restful environments (OSCORE) [24], DTLS with pre-shared keys among group members [25,26], attribute-based encryption [27], or even by relying on the information-centric networking (ICN) paradigm [28]. These solutions are concerned with the establishment of a secret key that is used for encrypting data [24,28] or the communication channels [25,26]. Such a key can be derived by a pre-shared symmetric key, a public key, or by the attributes of the communicating endpoints. Our solution is orthogonal to these approaches since our goal is to make sure that a client receives CoAP responses only from authorized servers.

Our work considers an SDN-based underlay architecture used instead of IP multicast for implementing group communication. Many related efforts are considering other alternatives to IP multicast, including the use of bit index explicit replication (BIRE) [29], MPL [30], and ICN [31]. We see these efforts complementary to our approach since our solution is agnostic to the underlying mechanism; therefore, it could be used with any of them.

Some related solutions use identity-based signature (IBS) to achieve similar goals with our work (see, for example, [32]). Although IBS removes the need for public keys, it introduces computational overhead, and it suffers from the so-called key escrow problem, since there is a single entity (the key generator) that knows all private keys. Our solution uses Ed25519 keys, which are 32 bytes long; hence, the gains, in terms of communication overhead of using an identity rather than an Ed25519 key, are small.

Our solution is designed for IoT devices and gateways that support the WoT specification. However, in recent years, a number of related technologies have emerged. For

example, under the umbrella of the *European IoT Platform Initiative* (https://iot-epi.eu/ accessed on 5 August 2021) a number of IoT gateway technologies were developed, by projects such as symbIoTe (https://iot-epi.eu/project/symbiote/ accessed on 5 August 2021), AGILE (https://iot-epi.eu/project/agile/ accessed on 5 August 2021), and Interiot (https://iot-epi.eu/project/inter-iot/ accessed on 5 August 2021). These efforts are now stalled. Since the only requirement of our solution is that device descriptions are encoded using JSON, we believe that it can be easily adapted for other gateway technologies.

Our system uses public keys for identifying IoT endpoints. A relevant technology that can be used instead is that of *decentralized identifiers* (DIDs) [33]. DIDs are closely related to LDPs. Additionally, when applied to the IoT, DIDs have some interesting security and privacy properties [34].

## 6. Conclusions

In this paper, we presented a security solution for managing group membership in IoT group communication. In particular, we leveraged linked-data proofs to assure that only valid group members can "advertise" their available resources. Our solution has intriguing security properties since it is a resilient event to private key breaches. Linked-data proofs allow the use of zero-knowledge proofs; our solution leverages this property in order to implement "selective disclosure" of available resources.

Future work in this area includes the integration of our solution with content confidentiality mechanisms (e.g., group OSCORE).

**Author Contributions:** Investigation, B.A. and N.F.; Project administration, B.A.; Software, N.F.; Writing—review & editing, B.A. and N.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia grant number 277.

**Data Availability Statement:** Not Applicable, the study does not report any data.

**Acknowledgments:** The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (227).

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

## References

1. Jiang, Y.; Miao, Y.; Alzahrani, B.; Barnawi, A.; Alotaibi, R.; Hu, L. Ultra Large-Scale Crowd Monitoring System Architecture and Design Issues. *IEEE Internet Things J.* **2021**, *8*, 10356–10366. [CrossRef]
2. Rahman, A.; Dijk, E. Group Communication for the Constrained Application Protocol (CoAP). 2014. Available online: https://www.hjp.at/doc/rfc/rfc7390.html (accessed on 5 August 2021).
3. Kaebish, S.; Kamiya, T.; McCool, M.; Charpenay, V.; Kovatsch, M. Web of Things Thing Description. 2020. Available online: https://www.w3.org/TR/wot-binding-templates/ (accessed on 5 August 2021).
4. Xia, W.; Wen, Y.; Foh, C.H.; Niyato, D.; Xie, H. A Survey on Software-Defined Networking. *IEEE Commun. Surv. Tutor.* **2015**, *17*, 27–51. [CrossRef]
5. Lara, A.; Kolasani, A.; Ramamurthy, B. Network Innovation using OpenFlow: A Survey. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 493–512. [CrossRef]
6. Bloom, B.H. Space/time trade-offs in hash coding with allowable errors. *Commun. ACM* **1970**, *13*, 422–426. [CrossRef]
7. Reed, M.J.; Al-Naday, M.; Thomos, N.; Trossen, D.; Petropoulos, G.; Spirou, S. Stateless multicast switching in software defined networks. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016; pp. 1–7. [CrossRef]
8. Shelby, Z.; Hartke, K.; Bormann, C. The Constrained Application Protocol (CoAP). 2014. Available online: https://iottestware.readthedocs.io/en/master/coap_rfc.html (accessed on 5 August 2021).
9. Kovatsch, M.; Matsukura, R.; Lagally, M.; Kawaguchi, T.; Toumura, K.; Kajimoto, K. Web of Things Architecture. 2020. Available online: https://www.w3.org/TR/wot-architecture/ (accessed on 5 August 2021).
10. Kellogg, G.; Champin, P.; Longley, D. JSON-LD 1.1. 2020. Available online: https://www.w3.org/TR/json-ld11/ (accessed on 5 August 2021).

11. Bernstein, D.J.; Duif, N.; Lange, T.; Schwabe, P.; Yang, B.Y. High-speed high-security signatures. *J. Cryptogr. Eng.* **2012**, *2*, 77–89. [CrossRef]
12. Josefsson, S. The Base16, Base32, and Base64 Data Encodings. 2006. Available online: https://www.hjp.at/doc/rfc/rfc4648.html (accessed on 5 August 2021).
13. Longley, D.; Sporny, M. Linked Data Proofs 1.0. 2021. Available online: https://w3c-ccg.github.io/ld-proofs/ (accessed on 5 August 2021).
14. Fotiou, N.; Mendrinos, D.; Polyzos, G.C. Edge-assisted traffic engineering and applications in the IoT. In Proceedings of the 2018 Workshop on Mobile Edge Communications (MECOMM'18), Budapest, Hungary, 20–25 August 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 37–42.
15. Alzahrani, B.; Fotiou, N. Enhancing Internet of Things Security using Software-Defined Networking. *J. Syst. Archit.* **2020**, *110*, 101779. [CrossRef]
16. Pfaff, B.; Pettit, J.; Koponen, T.; Jackson, E.; Zhou, A.; Rajahalme, J.; Gross, J.; Wang, A.; Stringer, J.; Shelar, P.; et al. The design and implementation of open vSwitch. In Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15), Oakland, CA, USA, 4–6 May 2015; USENIX Association: Oakland, CA, USA, 2015; pp. 117–130.
17. Gude, N.; Koponen, T.; Pettit, J.; Pfaff, B.; Casado, M.; McKeown, N.; Shenker, S. NOX: Towards an Operating System for Networks. *SIGCOMM Comput. Commun. Rev.* **2008**, *38*, 105–110. [CrossRef]
18. Lantz, B.; Heller, B.; McKeown, N. A network in a laptop: Rapid prototyping for software-defined networks. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, Monterey, CA, USA, 20–21 October 2010; ACM: New York, NY, USA, 2010; pp. 19:1–19:6.
19. Goldreich, O.; Oren, Y. Definitions and properties of zero-knowledge proof systems. *J. Cryptol.* **1994**, *7*, 1–32. [CrossRef]
20. Quisquater, J.J.; Quisquater, M.; Quisquater, M.; Quisquater, M.; Guillou, L.; Guillou, M.A.; Guillou, G.; Guillou, A.; Guillou, G.; Guillou, S. How to explain zero-knowledge protocols to your children. In Proceedings of the Conference on the Theory and Application of Cryptology, Santa Barbara, CA, USA, 20–24 August 1989; pp. 628–631.
21. Looker, T.; Steele, O. BBS+ Signatures 2020. 2020. Available online: https://w3c-ccg.github.io/ldp-bbs2020/ (accessed on 5 August 2021).
22. Sakemi, Y. (Ed.) Pairing-Friendly Curves v09. 2020. Available online: https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-pairing-friendly-curves-09 (accessed on 5 August 2021).
23. W3C. JSON-LD 1.1 Framing. 2020. Available online: https://www.w3.org/TR/json-ld11/ (accessed on 16 July 2020).
24. Tiloca, M.; Selander, G.; Palombini, F.; Park, J. Group OSCORE—Secure Group Communication for CoAP. 2021. Available online: https://www.ietf.org/id/draft-ietf-core-oscore-groupcomm-12.html (accessed on 5 August 2021).
25. Tiloca, M.; Nikitin, K.; Raza, S. Axiom: DTLS-Based Secure IoT Group Communication. *ACM Trans. Embed. Comput. Syst.* **2017**, *16*, 1–29. [CrossRef]
26. Park, C.S. Security Architecture for Secure Multicast CoAP Applications. *IEEE Internet Things J.* **2020**, *7*, 3441–3452. [CrossRef]
27. Basu, S.S.; Tripathy, S. Securing Multicast Group Communication in IoT-Enabled Systems. *IETE Tech. Rev.* **2019**, *36*, 83–93. [CrossRef]
28. Gündoğan, C.; Amsüss, C.; Schmidt, T.C.; Wählisch, M. IoT content object security with OSCORE and NDN: A first experimental comparison. In Proceedings of the 2020 IFIP Networking Conference (Networking), Paris, France, 22–26 June 2020; pp. 19–27.
29. Wijnands, I.; Rosen, E.C.; Aldrin, S.; Przygienda, T.; Dolganow, A. Multicast Using Bit Index Explicit Replication (BIER). 2017. Available online: https://www.hjp.at/doc/rfc/rfc8279.html (accessed on 5 August 2021).
30. Hui, J.; Kelsey, R. M Multicast Protocol for Low-Power and Lossy Networks (MPL). 2016. Available online: https://tex2e.github.io/rfc-translater/html/rfc7731.html (accessed on 5 August 2021).
31. Gündoğan, C.; Amsüss, C.; Schmidt, T.C.; Wählisch, M. Toward a restful information-centric web of things: A deeper look at data orientation in CoAP. In Proceedings of the 7th ACM Conference on Information-Centric Networking (ICN'20), Virtual, 29 September–1 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 77–88.
32. Felde, N.; Grundner-Culemann, S.; Guggemos, T. Authentication in dynamic groups using identity-based signatures. In Proceedings of the 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), Limassol, Cyprus, 15–17 October 2018; pp. 1–6. [CrossRef]
33. W3C Credentials Community Group. A Primer for Decentralized Identifiers. 2019. Available online: https://w3c-ccg.github.io/did-primer/ (accessed on 5 August 2021).
34. Kortesniemi, Y.; Lagutin, D.; Elo, T.; Fotiou, N. Improving the Privacy of IoT with Decentralised Identifiers (DIDs). *J. Comp. Netw. Commun.* **2019**, *2019*, 8706760:1–8706760:10. [CrossRef]