Internet Access for LoRaWAN Devices Considering Security Issues

Ramon Sanchez-Iborra, Jesús Sánchez-Gómez, Salvador Pérez, Pedro J. Fernández,
José Santa, José L. Hernández-Ramos, Antonio F. Skarmeta
Department of Information and Communications Engineering
University of Murcia
Murcia, Spain
Email: {ramonsanchez, jesus.sanchez4, salvador.p.f, pedroj, josesanta, jluis.hernandez, skarmeta}@um.es

***Abstract*—Low-Power Wide-Area Networks (LP-WAN) are becoming the cornerstone for IoT connectivity these days, given the long range coverage and the number of devices that could be interconnected through a single gateway. Among LP-WAN proposals, LoRaWAN (Long Range Wide Area Network) is receiving special attention, because of its adaptability to actual transmission and scenario conditions and its license-free operation. However, current LoRaWAN specifications especially address physical and link-level communication requirements, leaving network and higher network protocols aside. Moreover, security mechanisms currently considered suffer from lack of flexibility, requiring the pre-sharing of cryptographic material. In the Future Internet, IPv6 support for end-to-end thing connectivity is essential, and this entails a challenge for this kind of constrained networks. In this paper, we address the connectivity issue through an implementation of an IPv6 adaptation mechanism that is being defined at the IETF and we explore the security flaws that could be solved by the integration of an Internet Key Exchange (IKE)-flavoured solution especially adapted to constrained communication technologies. The security features proposed intend to cover the negotiation of key material for LP-WAN technologies above the link layer, avoiding manual and insecure configurations.**

***Keywords*-LP-WAN; LoRaWAN; IPv6; SCHC; Security; ED-HOC; Internet**

I. Introduction

The Internet of Things came with a new set of low-power communication technologies that usually requires nearby proxies, such as ZigBee or Bluetooth LE. New proposals in this line increase the communication range to access constrained network nodes at the price of reducing data rate, which is not an issue for these IoT applications most of the times. Technologies following this approach have been coined as Low-Power Wide-Area Networks (LP-WAN), attracting great attention due to its beneficial characteristics of long coverage ranges, low energy consumption and high scalability [1]. One of the most prominent LP-WAN solutions is LoRaWAN (Long Range Wide Area Network), due to its flexibility and high-grade of adaptability to the user needs [2]. Unlike other technologies, such as Sigfox, LoRaWAN is not subject to operation licenses, and both end-devices and proxies can be purchased and tested following local frequency regulations.

However, a notable shortcoming in LP-WAN technologies is the lack of a common ground for ensuring interoperability with the current and future Internet, allowing the full interconnection of LP-WAN-powered things. In this line, IPv6 has been adopted as a nexus to interconnect IoT devices with the Internet services, as we can see in other short-range technologies implementing the IEEE 802.15.4 specifications, such as IPv6 over Low power Wireless Personal Area Networks (6LoWPAN); hence, the provision of such capability for the case of LP-WAN could be essential for its definitive implantation in future IoT scenarios. However, the main issue found here is the highly constrained feature of end-nodes: limited bit-rate, reduced packet size, etc. These constrains are even more severe than for the case of 802.15.4 networks. For this reason, the straight integration of IPv6 packets into LP-WAN frames is not trivial, requiring extreme compression mechanisms.

Moreover, it is important to revisit security features included in LP-WAN technologies, especially when Internet reachability is aimed for these devices. Given the constrained nature of these technologies, some security features could be left to higher layers or require manual configurations. For the case of LoRaWAN, which is the technology explored in this work, manual intervention is needed to insert cryptographic material in end-devices, which leads to administrative complexities and a source of potential cyber-security risks. Concretely, the security mechanisms employed in LoRaWAN are based on a pre-shared key that should be hard-coded in each end-device and shared with the service in charge of handling the application payload. This strategy lacks of flexibility regarding the key update process when the pre-shared key or the subsequent ones generated from it are compromised; besides, the join procedure used by end-nodes in order to gain access to the network is prone to Denial of Service (DoS) attacks as discussed in following sections [3].

In this line, this work focuses on providing IPv6 connectivity to LP-WAN nodes enabled with LoRaWAN technology, but especially considering security issues. To this end, an additional IPv6 adaptation layer has been developed, in charge of packet translation tasks (compression/decompression), which has been designed to operate at both the end-node and proxy sides; thereby, bidirectional flows can be established between LoRaWAN end-nodes and

IPv6 nodes within an external administrative domain or the Internet. To pave the way of this LP-WAN Internet access, the current security features of LoRaWAN are enhanced with the integration of an Internet Key Exchange (IKE)-like approach that automatizes the distribution of cryptographic material and can even improve the overall performance of the network. Therefore, the main contributions of this work are the following: (i) an overview of the problem of integrating LP-WANs within the Internet, (ii) a proposed implementation of the IPv6 adaptation layer, and (iii) a description of a flexible security extension for LP-WANs.

The rest of the paper is organized as follows. Section II places the work in the scientific literature. Section III introduces LoRa, LoRaWAN, and its security mechanisms. Section IV presents the IPv6-supportive architecture for LoRaWAN, while Section V focuses on the security issues present in the current security features of LoRaWAN and explores potential extensions. Finally, Section VI concludes this work, presenting the most important findings.

II. Related Work

Due to the interest devoted to the challenge of allowing IPv6 communications on LP-WAN links, the Internet Engineering Task Force (IETF) formed the IPv6 over Low Power Wide-Area Networks (lpwan) working group in 2016. The advances proposed by this group also consider the use of higher-layer protocols like User Datagram Protocol (UDP), and Constrained Application Protocol (CoAP) over LP-WAN networks. Two active Internet drafts are currently available addressing these issues [4], [5]. These documents propose the Static Context Header Compression (SCHC) for enabling the adaptation of IPv6/UDP/CoAP headers to the stringent requirements posed by LP-WAN systems. A detailed exploration of these mechanisms will be presented in the following sections.

From the academia perspective, very few works have addressed the integration of higher-layer protocols within LP-WAN architectures [6], [7], [8]. The work in [6] presented two different mechanisms for adapting the IPv6 header to fit it within the LoRaWAN Maximum Transfer Unit (MTU), one based on a static transformation agreement between the end-node and the gateway, and a dynamic one implying a previous negotiation. However, these mechanisms are based on an IPv6 over 6LoWPAN adaptation approach, which implies a network link with higher data-rates and shorter ranges than LP-WAN networks. The authors of [7] proposed a similar approach, by encapsulating IPv6 packets over LoRa systems by replacing the IEEE 802.15.4 support of Contiki, in a 6LoWPAN case of study, with LoRa. Again, the same problem of protocol overhead arises when using a compression algorithm not adapted to LP-WAN, but in this case the LoRaWAN layer is also removed. Finally, the work in [8] followed the IETF lpwan guidelines, implementing the SCHC compression mechanisms and even proposing an improvement to reduce memory and processing demands in constrained devices. Nevertheless, the LP-WAN performance tests were done theoretically by using an time-on-air calculator.

Regarding security issues in LP-WAN systems, some works have recently explored the potential vulnerabilities of the security mechanisms adopted by LoRaWAN [3], [9]. The work in [3] focused on evaluating the security of the LoRAWAN join process with special emphasis on the nonce generation during this phase. Authors found this scheme to be prone to DoS attacks by the regeneration of an already used nonce by an end-device. Jamming attacks were also considered and the system robustness was evaluated, showing its vulnerability against this kind of malicious events. In turn, the work in [9] presented a more descriptive analysis of the security risks suffered by LoRaWAN networks. Authors found that the long LoRaWAN packet's time-on-air is highly attractive to potential attackers because it facilitates jamming, replay or wormhole attacks. A physical attack was also performed to an end-node, extracting the pre-shared key from it, hence, leaving the device open to hijacking. However, these works did not propose any concrete security scheme in order to face the cited weaknesses.

In this work, a generic IPv6 over LoRaWAN solution is proposed, providing a deep discussion of its main features and potential applications. This base architecture is complemented with a security extension based on the Ephemeral Diffie-Hellman Over COSE (EDHOC) protocol [10], which makes easier the configuration of nodes avoiding the manual distribution of keys, and even improves the performance when key updates are necessary by the end-device, as compared with current security features of LoRaWAN.

III. LoRaWAN

LoRa is based on a proprietary Chirp Spread Spectrum (CSS) modulation by Semtech. This modulation scheme permits to achieve very long transmission distances of more than 10 km at the expense of reducing its communication capabilities in terms of data rate and packet length. LoRa presents three different tunable parameters that permits it to be adapted to different scenarios. These parameters are: Spreading Factor (SF), Coding Rate (CR), and Bandwidth (BW). These factors determine the aforementioned trade-off between the achievable distance and the transmission features (throughput, payload size, etc.). While LoRa defines the physical (PHY) layer of the communication stack, LoRaWAN defines the Medium Access Layer (MAC), introducing interesting characteristics such as acknowledgement schemes, security suites, or synchronization tools, among others. For further insides about LoRaWAN settings, please refer to [2].

Due to the need of employing low-bit rates for achieving long coverage ranges and robust links, the transmission

times-on-air are notably long; for that reason, it is highly valued to reduce the packets size in order to avoid undesirable effects over the transmissions, e.g., interferences, collisions, or excessive channel occupation. This is the main challenge for introducing higher layer protocols inside LoRaWAN frames. Acceptable payload lengths range from 1 to 242 bytes, hence, please observe that higher layer's Protocol Data Units (PDU) length should be as reduced as possible. The mechanisms proposed by the IETF standardization body for achieving a proper compression of the network and above layers are explored in the next section.

From a security perspective, LoRaWAN implements a cryptographic mechanism based on AES-128 working in counter (CTR) mode. It makes use of a pre-shared key from which two additional keys are derived for securing the session. Depending on the adopted join scheme, some of these keys should be manually hard-corded on the devices, so a manual intervention is needed for each single device composing the network. Besides, it is not defined any update period for these keys, which might be identified as a security vulnerability as it will be examined in next sections. Therefore, it can be seen that the security solution employed in LoRaWAN lacks of flexibility in order to quickly react against security attacks and the management efforts are considerably high as manual intervention is needed for its configuration.

IV. Adding IPv6 Support to LP-WAN

In this section, the IPv6 transport over LP-WAN is presented, including an overview of the header compression process under consideration.

A. General Encapsulation Proposal

The followed approach for the encapsulation process is shown in Fig. 1. As can be seen, application-level packets are encapsulated in UDP datagrams. It is assumed this protocol in order to diminish the network overload over the constrained LP-WAN link and because connection-oriented services are not common in these networks. Then, regular IPv6 datagrams are adapted to be sent over the given LP-WAN technology. For this, the SCHC algorithm [4] is used. The final compressed IPv6 packet is sent as the payload of the physical frame transmitted through the LP-WAN link.

As indicated in Fig. 1, the previous adaptation is reverted upon the reception of the frame in network gateway. The original IPv6 packet is then forwarded through a regular link, probably traversing the Internet, until reaching the final IPv6 destination node, which implements the application level of the concrete service. Of course, this process is bi-directional and the end-nodes are able to receive and de-compress IPv6 compressed packets as well.

B. Header Compression Process

As aforementioned, the Static Context Header Compression (SCHC) proposed in [4] by the IETF standardization

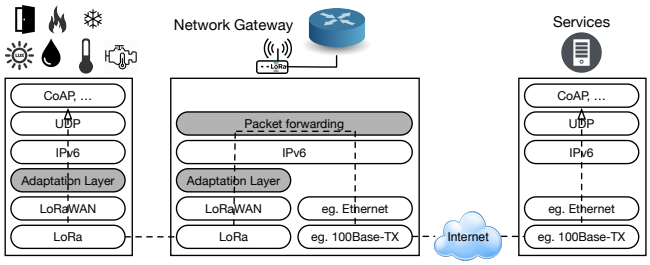

Figure 1. IPv6 encapsulation over LP-WAN

body has been adopted. This scheme is a header compression strategy and fragmentation functionality for IPv6 over low data-rate networks such as LP-WAN. It is based on a common static context that is stored in every end-device and the Internet gateway, enabling them to perform the translation between the full-length IPv6 packets and their reduced version. The context provides information describing the content of the different fields of the packet headers and it is pre-provisioned for both extremes of the communication. Consequently, this context might not be transmitted over the LP-WAN link, e.g., it could be exchanged via an off-line method, and it must not change during the communication. This avoids resynchronization procedures and ensures the consistency of the (de)compression process.

The context is composed of a set of rules that defines how to compress the different fields of the IPv6, ICMPv6, and UDP headers. The idea behind the SCHC compression is that LP-WAN traffic is very predictable. Thus, each rule is identified by a Rule-ID that represents an individual flow to/from a given device so, instead of sending the entire field values, just the Rule-ID is included in every compressed packet within each corresponding flow. The Rule-ID represents the rule that better matches the header fields and yields the greatest header compression. Consequently, every rule must also define the decompression actions to generate a valid IPv6 packet upon reception.

In this work, the rules proposed in [11] have been included in the considered static context for the ICMPv6 protocol. In addition, a rule for managing IPv6/UDP traffic has been added to the static context of our implementation, too. This rule is bidirectional and it is the only one that applies to all UDP traffic. It is defined as follows: (i) the source and destination ports are sent, and (ii) the length and checksum fields are suppressed. Observe that the length and checksum values can be independently computed in both extremes and the integrity of the packet can be assured by lower layers mechanisms. As a result, the compressed size of the UDP header is just four bytes. In this work, fragmentation is not considered because all the sent packages over the network have a predefined size smaller than the maximum MTU of the adopted LP-WAN solution, namely, LoRaWAN.

C. IPv6 over LoRaWAN Solution

As stated above, the implemented solution for encapsulating IPv6 packets within LoRaWAN frames was developed following the guidelines in [4], [11], and taking as a starting point the software developed by the authors of [11][1]. This implementation clearly differentiates two possible directions of traffic, namely, uplink and downlink traffic. For that reason, depending on the traffic direction, the actions needed to forward the packets are different.

In an uplink communication, the end-device generates an IPv6 datagram, which is reduced following the SCHC compression. Then, the resulting packet is sent over the LoRaWAN link. Note that the SCHC compression scheme does not modify the packets payload, since it only reduces the information in the IPv6/UDP or IPv6/ICMPv6 headers and introduces the corresponding Rule-ID in a specific header field. When the uplink packet is received at the network gateway, it stores the end-devices source address in its neighbor devices list. Thus, this list contains the addresses of those end-nodes that accessed the external network though the network gateway. Then, following the instructions established by the corresponding Rule-ID, the received packet is decompressed resulting in the original IPv6 packet. If the SCHC decompression procedure fails to decode the information, the packet is discarded. Finally, if the decompression is successful, the resulting packet is sent over the external network interface of the network gateway.

When the network gateway receives a downlink packet coming from its external network interface, e.g., Ethernet, the reverse steps take place and the packet is compressed using the SCHC compression procedure. The resulting packet is sent over the LoRaWAN link and, finally, the end-node reconstructs the original packet in order to pass it up to the application layer through the complete IPv6 stack.

V. LoRaWAN Security Analysis

A. LoRaWAN Security Procedures

As stated above, the security implemented in off-the-shelf LoRaWAN systems is based on a pre-shared key, so-called Application Key (AppKey), from which two additional keys are derived: the Network Session Key (NwkSKey) and the Application Session Key (AppSKey). The latter is employed to protect the packet payload, so it is shared just between both extremes of the communication in order to ensure the confidentiality of these data. In turn, the NwkSKey is shared among the end-device and the networking elements aiming to permit routing tasks and sanity checks. The establishment of such keys depends on the selected join scheme, which permits end-nodes to be authenticated by the network.

Two different security procedures have been defined in LoRaWAN, namely, Over-the-air Activation (OTAA) and Activation by Personalization (ABP). During the OTAA, which is the usually recommended scheme, a message exchange including cryptographic material is performed between the end-device and the network server. By using this material and the AppKey, both the NwkSKey and the AppSKey are derived. After this step, communications are secured by using the derived keys. This process is carried out when the node connects for the first time to the network or in case of connectivity loss, however, it is not defined any update period.

Even more static, the ABP scheme proposes to hard-code the two session keys, so the update process needs manual intervention from the network manager. Therefore, these devices do not perform any periodic update of such keys (static session keys), which means that the probability of the NwkSKey and the AppSKey being compromised is greater since they are constantly used to protect the communications. Thus, if an attacker successes in obtaining these session keys, all the encrypted information exchanged between the end-device and the network server will be accessible during a long period of time until the attack is detected.

B. Security Alternatives

Cybersecurity problems caused by the use of static NwkSKey and AppSKey motivate the need to incorporate a session key update mechanism into LoRaWAN to ensure an adequate protection of the information shared between the end-device and the network server. In addition, such key update mechanism should be periodically executed according to different practical aspects of the network, e.g., the degree of sensitivity of the exchanged information. In this sense, there are different protocols that may provide this functionality such as Datagram Transport Layer Security (DTLS) handshake [12], Internet Key Exchange (IKE) v2 [13] and EDHOC [10].

DTLS handshake allows to establish security parameters between client and server, e.g., cryptographic algorithms or authentication schemes, so that these devices are able to calculate shared symmetric keys in order to protect the communications between them. In case of incorporating this protocol into LoRaWAN for the NwkSKey and AppSKey update, the end-device and the network server should select a cipher-suite that fits into the security mechanisms already established in this network. Particularly, the cipher-suite TLS_PSK_DHE_WITH_AES_128_CCM_8 allows setting the use of the AppKey to carry out an authentication process based on a pre-shared key (PSK). In addition, by using the Diffie-Hellman algorithm (DHE), the devices can generate as many session keys as necessary and independent of each other. This way, if one of these keys is compromised, encrypted information with previous keys is kept protected against unauthorized access, achieving the Perfect Forward Secrecy (PFS) [14]. However, it should be pointed out that the choice of DTLS handshake as mechanism for updating

[1]https://github.com/tlagos1/LoRA_IPv6_implementation

session keys requires to exchange, at least, six messages between the end-device and the network server, according to the DTLS specification. Therefore, if the LoRaWAN network is made up by a large number of end-devices, the use of this handshake protocol can lead to a high traffic overhead.

As LoRaWAN communications rely on IPv6 packets in our proposed scheme, IPSec can also be used to protect the end-to-end communications, using IKEv2 as key-management service in order to generate and frequently refresh keying material between client and server. In this case, the authentication can be preformed using a pre-shared key as well as certificates. IKEv2 mechanisms make use of the Diffie-Hellman algorithm, so it also achieves PFS protection as well as DTLS does. The Diffie-Hellman algorithm performance cost depends on the chosen cipher-suite. For this type of strongly constrained scenarios, an Elliptic Curve Group [15] can be selected to reduce that cost and processing time. In addition, IKEv2 uses four messages to perform the initial security negotiation, and two messages for each keying update. Nevertheless, the big inconvenient in IKEv2 protocol is its message sizes that exceed the supported limit size of LoRaWAN messages, even at lower data-rates. This limit is located at 51 bytes, which is enough to make IKEv2 non-recommendable for this kind of scenarios.

As seen, both DTLS handshake and IKEv2 present problems to carry out the update of the NwkSKey and the AppSKey in LoRaWAN, as they were not designed to work in this highly constrained scenarios. Accordingly, we propose the use of EDHOC to cope with this need.

C. EDHOC-based Proposal

EDHOC is a lightweight protocol that allows the establishment of shared symmetric keys between client and server. It is based on an authenticated Elliptic Curve Diffie-Hellman algorithm (ECDH) [16] with the purpose of providing PFS and confirming the identity of the involved devices. In this sense, EDHOC specifies two authentication modes, in particular, authentication based on public key, i.e., raw public keys and certificates, and authentication based on a pre-shared key. In order to incorporate this protocol into LoRaWAN, we select the second authentication mode so that the AppKey acts as pre-shared key. Besides, EDHOC only requires the exchange of three messages between the end-device and the network server to compute new session keys, which entails a lower network overload than DTLS handshake protocol and initial IKEv2 exchanges. In addition, EDHOC messages are protected by CBOR Object Signing and Encryption (COSE) [17], and are encoded following the Concise Binary Object Representation (CBOR) [18]. This aspect implies that the size of such messages is reduced, favoring their processing by constrained end-devices in terms of memory, storage and energy consumption.

In order to demonstrate and show the advantages offered by EDHOC as method of updating the NwkSKey and AppSKey keys in LoRaWAN, we are currently developing a first prototype of the IPv6/UDP stack explained above incorporating this protocol. In particular, our implementation of EDHOC uses the elliptic curve P-256 during the Diffie-Hellman algorithm, since this curve already provides the minimum security level recommended by the NIST, that is, 128-bit [19]. For the same reason, we selected AES with 128-bit symmetric keys as an authenticated encryption algorithm (AEAD) [20]. This algorithm, in addition to providing confidentiality to the information exchanged between the end-device and the network server, allows verifying its integrity and authenticity. Furthermore, EDHOC establishes the use of a key derivation function (KDF) to compute the symmetric key shared between client and server from the secret obtained by the Diffie-Hellman algorithm. The main goal of this function is to destroy existing algebraic relationships between different public keys that can lead to obtain the same secret. This way, new computed symmetric keys are cryptographically strong [21]. In our case, we choose the HKDF function [22] since it is recommended by NIST as a mechanism for key derivation through extraction-then-expansion [23] and, additionally, EDHOC specification forces its implementation. Finally, regarding the COSE and CBOR libraries used in our development, we use the GitHub project[2] pointed out in COSE specification. Taking into account the values established for EDHOC security parameters described above, part of our future work focuses on the evaluation of the developed prototype working over a SCHC-based IPv6 over LoRaWAN architecture.

VI. Conclusion

In this work, a novel IPv6-based architecture over LP-WANs considering security issues has been presented. Concretely, the LoRaWAN technology has been adopted as it is one of the novel IoT communication technologies that is receiving more attention during the last times.

The potential of this type of long-range transmission technologies is evident, however some shortcomings have been identified in the fields of interoperability and security. Focusing on the former, in order to achieve effective end-to-end communications between end-devices using different IoT radio access technologies and belonging to different administrative domains, a common ground is needed. Thus, IPv6 has been recognized as a valuable nexus for enabling a general interoperability among the different networks composing the heterogeneous IoT ecosystem. However, IPv6 introduces a great overhead that is not assumable by LP-WAN-based networks. For that reason, the SCHC compression scheme, currently under study by the IETF, has been adopted as a solution for reducing the IPv6 and UDP headers. In this

[2]https://github.com/cose-wg/COSE-C

line, the full LoRaWAN stack including these protocols has been deeply explored.

Regarding security, we have identified that the procedures used by LoRaWAN lack from flexibility in realistic deployments. Hence, some protocols for the establishment and auto-update of shared keys have been explored. The pros and cons of well-known key exchange schemes have been studied for the case of LoRaWAN networks, concluding that EDHOC might be an interesting option for this kind of systems by adopting a proper security suite. As future work, we plan to continue developing our proposed framework in order to evaluate the performance of the LoRaWAN system and gather results that allow us to compare our proposal with currently available security procedures.

ACKNOWLEDGMENT

This work has been sponsored by the Spanish Ministry of Economy and Competitiveness, through the PERSEIDES (ref. TIN2017-86885-R) and USEIT (ref. PCIN-2016-010) projects.

REFERENCES

[1] R. Sanchez-Iborra and M.-D. Cano, "State of the art in lp-wan solutions for industrial iot services," *Sensors*, vol. 16, no. 5, 2016.

[2] "A technical overview of LoRa and LoRaWAN," LoRa-Alliance, Tech. Rep., 2015.

[3] S. Tomasin, S. Zulian, and L. Vangelista, "Security analysis of LoRaWAN join procedure for Internet of Things networks," in *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. IEEE, mar 2017, pp. 1–6.

[4] A. Minaburo, L. Toutain, and C. Gomez, "LPWAN Static Context Header Compression (SCHC) and fragmentation for IPv6 and UDP," RFC Draft, Internet Engineering Task Force, Oct. 2017. [Online]. Available: https://tools.ietf.org/html/draft-ietf-lpwan-ipv6-static-context-hc-07

[5] A. Minaburo and L. Toutain, "LPWAN Static Context Header Compression (SCHC) for CoAP," RFC Draft, Internet Engineering Task Force, Sep. 2017. [Online]. Available: https://tools.ietf.org/html/draft-ietf-lpwan-coap-static-context-hc-02

[6] P. Weber, D. Jckle, D. Rahusen, and A. Sikora, "IPv6 over LoRaWAN," in *2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, Sept 2016, pp. 75–79.

[7] S. Thielemans, M. Bezunartea, and K. Steenhaut, "Establishing transparent IPv6 communication on lora based low power wide area networks (LPWANs)," in *2017 Wireless Telecommunications Symposium (WTS)*, April 2017, pp. 1–6.

[8] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "LSCHC: layered static context header compression for lpwans," *CoRR*, vol. abs/1708.05209, 2017.

[9] E. Aras, G. S. Ramachandran, P. Lawrence, and D. Hughes, "Exploring the security vulnerabilities of LoRa," in *3rd IEEE International Conference on Cybernetics (CYBCONF)*. IEEE, jun, pp. 1–6.

[10] G. Selander, J. Mattsson, and F. Palombini, "Ephemeral diffie-hellman over cose (edhoc)," RFC Draft, July 2017. [Online]. Available: http://www.ietf.org/internet-drafts/draft-selander-ace-cose-ecdhe-07.txt

[11] T. Lagos and D. Dujovne, "LPWAN Static Context Header Compression (SCHC) for ICMPv6," RFC Draft, Internet Engineering Task Force, Jun. 2017. [Online]. Available: https://tools.ietf.org/html/draft-lagos-lpwan-icmpv6-static-context-hc-00

[12] E. Rescorla and N. Modadugu, "Datagram transport layer security version 1.2," RFC 6347, January 2012. [Online]. Available: http://www.rfc-editor.org/rfc/rfc6347.txt

[13] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen, "Internet key exchange protocol version 2 (ikev2)," RFC 5996, September 2010. [Online]. Available: https://www.rfc-editor.org/info/rfc5996

[14] H. Krawczyk, "Perfect forward secrecy," in *Encyclopedia of Cryptography and Security*. Springer, 2011, pp. 921–922.

[15] D. Fu and J. Solinas, "Elliptic curve groups modulo a prime (ECP groups) for IKE and IKEv2," RFC 5903, June 2010. [Online]. Available: https://www.rfc-editor.org/info/rfc5903

[16] D. McGrew, K. Igoe, and M. Salter, "Fundamental elliptic curve cryptography algorithms," RFC Editor, RFC 6090, February 2011, http://www.rfc-editor.org/rfc/rfc6090.txt.

[17] J. Schaad, "CBOR object signing and encryption (COSE)," RFC 8152, July 2017. [Online]. Available: https://www.rfc-editor.org/info/rfc8152

[18] C. Bormann and P. Hoffman, "Concise binary object representation (CBOR)," RFC 7049, October 2013. [Online]. Available: https://www.rfc-editor.org/info/rfc7049

[19] E. Barker, "Recommendation for key management part 1: General (revision 4)," National Institute of Standards and Technology, Tech. Rep. 800-57, 1 2016.

[20] D. McGrew, "An interface and algorithms for authenticated encryption," RFC 5116, January 2008. [Online]. Available: https://www.rfc-editor.org/info/rfc5116

[21] A. Langley, M. Hamburg, and S. Turner, "Elliptic curves for security," RFC 7748, January 2016. [Online]. Available: https://www.rfc-editor.org/info/rfc7748

[22] H. Krawczyk and P. Eronen, "HMAC-based extract-and-expand key derivation function (HKDF)," RFC 5869, May 2010. [Online]. Available: https://www.rfc-editor.org/info/rfc5869

[23] L. Chen, "Recommendation for key derivation through extraction-then-expansion," National Institute of Standards and Technology, Tech. Rep. 800-56C, 11 2011.