Optimizing Energy Efficiency of Secured IoT Communication by OpenHip

Peter Kaňuch, Dominik Macko
Faculty of Informatics and Information Technologies
Slovak University of Technology
Bratislava, Slovakia
Email: xkanuch@stuba.sk, dominik.macko@stuba.sk

***Abstract*—The research in the area of Internet-of-Things (IoT) security is still underway due to the growing IoT networks. The main goal of the existing works, optimizing security protocols, is to make them more efficient in order to reduce their energy requirements. The result is the extended lifetime of the device powered by a battery, while preserving all the security features of the protocol such as confidentiality, integrity, authenticity, etc. Based on the analysis, we have decided to focus on the protocol HIP (Host Identity Protocol), identified several optimization possibilities for efficient use in the IoT area, and proposed its modification. The analytical and experimental evaluation shows that the proposed modifications are beneficial regarding the energy efficiency of the HIP protocol.**

***Keywords*—energy efficiency; Internet of Things; low-power communication; security; wireless sensor networks**

I. Introduction

More and more interconnected devices are now being connected to the Internet, referred to as the Internet of Things (IoT) [1]. Since the number of interconnected IoT devices in the world grows rapidly (used in industry, smart cities, agriculture, etc. [2]), they gained the attention of network attackers. Therefore, IoT security is the most crucial and we must think about security features, such as authenticity, integrity, confidentiality, policy [3]–[5]. However, it is not easy for IoT devices to offer strong security features since many IoT devices are constrained on power and resources side. Most of the IoT sensor nodes are powered by batteries or harvesting energy from the environment. Therefore, optimization of traditionally used security protocols for the usage in IoT is a must. There already exist multiple research works in this area (e.g. [6]–[8]); however, there is still a space for optimization of communication energy efficiency.

In this paper, we focus on the HIP protocol [9], which was developed for a key-exchange procedure. We have identified several possible optimizations of its energy efficiency and optimized this protocol to efficiently secure the IoT communication while keeping the security level offered by the original protocol.

This work was partially supported by the Slovak Research and Development Agency(APVV-15-0789), the Slovak Cultural and Educational Grant Agency (KEGA 011STU-4/2017), the Slovak Scientific Grant Agency (VEGA 1/0836/16), the project "University Science Park of STU Bratislava" (ITMS 26240220084), co-funded by the European Regional Development Fund, and the project no. 2018/14427:1-26C0.

The organization of the paper follows this structure: In the next section, the works related to energy-efficient IoT security protocols are summarized and analyzed. In Section III, the proposed HIP modifications, optimizing its energy efficiency, are described in more detail. Section IV outlines validation of the selected modifications by their implementation into the OpenHIP library [10]. And finally, the conclusions of this work are provided in Section V.

II. Related Works

There are several existing solutions of energy-efficient communication security [11], optimized for usage in the IoT area.

Protocol DTLS (Datagram Transport Layer Security) [12] supports UDP communication, which is more energy efficient than TCP due to the transport-protocol header size. There was also proposed its modification eeDTLS (Energy-Efficient DTLS) [6], which reduced protocol headers and optimized the handshake process. Lithe (Lightweight Secure CoAP - Constrained Application Protocol - for the IoT) [7] is also one of the existing DTLS optimizations, which uses a combination of DTLS and CoAP to provide security. The proposed optimization consists especially of headers compression.

The authentication and data-encryption security features are not supported directly in the CoAP protocol, which is one of the most widely used IoT application protocols. Therefore, IPSec (Internet Protocol Security) is also used in the area of IoT [13], or its energy-efficient modification LKA (Lightweight Key Agreement) [8]. It is a minimized configuration of the IKEv2 (Internet Key Exchange) protocol that offers basic options only, such as the usage of a single cryptographic algorithm.

Protocol HIP [9] is designed for key exchange, as an alternative to IKE of the IPSec, which separates the identification (i.e. cryptographic identifiers) of devices from their locations (i.e. IP addresses). Such a security feature, enabling anonymous locations and supporting mobility, is very useful for many IoT applications. There are multiple works targeting the HIP protocol and making it more energy efficient. For example, the HIP-TEX modification [14] integrated a distribution mechanism into the HIP to cope with limited resources of IoT devices. Another modification, HIP-DEX [15], used the Elliptic-curve cryptography to lighten the computational requirements of the

This is an accepted version of the paper:
P. Kaňuch and D. Macko, "Optimizing energy efficiency of secured IoT communication by OpenHip," in 2019 42nd International Conference on Telecommunications and Signal Processing (TSP), 2019, pp. 174-177.
doi: 10.1109/TSP.2019.8769096
URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8769096

protocol. This solution was further optimized by the Slimfit modification [16], which introduced a compression into the HIP header reducing the fragmentation rate. A combination of compression and distribution mechanisms was proposed in CD-HIP [17] to optimize the HIP for usage in constrained IoT devices using 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) communication.

III. The Proposed HIP Optimizations

The analyzed existing related works revealed that the optimization of the standard protocols for usage in constrained IoT devices and networks is quite common. We have selected the HIP protocol for our interests due to its unique benefits of hiding locations and mobility support, which are often required in healthcare or military industry. In our work, we propose six small modifications of the HIP protocol, which are further described in the following subsections.

A. Removal of the CloseAck Message and the Temporary Closing State

There are two HIP messages used for termination of a connection between the devices (see Fig. 1):

- Close - signals the end of the connection,
- CloseAck - confirms the end of the connection between the communicating devices.

We propose to remove the CloseAck message. Such a modification enables us to also remove the state (in the protocol state machine) between the messages when the device is waiting for the confirmation message, labeled as a temporary closing state.

This proposal saves the processor time for execution of instructions intended for message processing and the time in which the device is only waiting for the confirmation message.

Such a solution is intended for the IoT area, where the server is powered directly from a grid (i.e. an ”unlimited” energy source) and the IoT device is powered by a battery (i.e. a limited energy source). The following situations can occur:

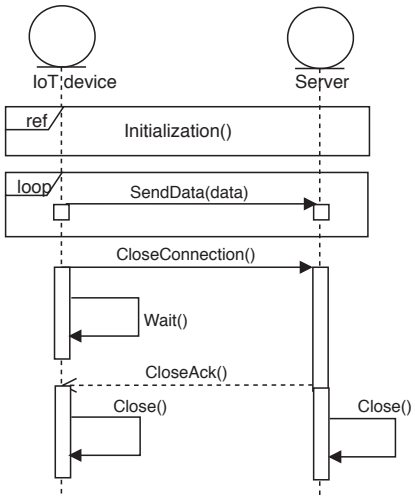

Fig. 1. The communication process of the HIP protocol.

1) The IoT device terminates the connection by sending the Close message, afterward, the server ends the connection by receiving this message. Also, the reverse situation can occur.
2) The IoT device terminates the connection, but the server does not receive the message due to a message loss. In such a case, the server waits for a time-out timer and ends the connection upon its expiration.

Due to the second situation, we have to state a precondition: only an IoT device can terminate the connection, because waiting for a timer expiration on the IoT device is not energy efficient.

B. Reduction of the Parameter Format

During the association of communicating devices, there are four types of messages exchanged: I1, R1, I2, and R2.

The R1 and I2 messages carry multiple parameters (e.g. the R1 contains: Puzzle, DH-R, HI-R, HIP, Transform, ESP Transforms, Echo Request, SIG). All the messages include information about a type and a length. In our work, we propose to remove the type and length fields of parameters. For this purpose, it is necessary to define a fixed order of the parameters. A difference between the packets before and after the proposed modification is shown below.

```
HIP Parameters Before:
ESP_INFO (type=65. length=12)
Reserved: 0x0000
...
R1_COUNTER (type=321. length=20)
Reserved: 0x00000000
...
************************************
HIP Parameters After:
ESP_INFO
Reserved: 0x0000
...
R1_COUNTER
Reserved: 0x00000000
...
```

Using this modification, it is possible to reduce the sending message size by 32B. In the prototype solution, we have modified the I2 message as a proof-of-concept and saved 16B.

C. Removal of the HI-R Parameter

The next partial proposal of the modification is to remove the HI-R (Host Identity - Responder) parameter, which is used as a public key in the cryptographic algorithms, such as RSA, DSA, etc. By its removing, it is also possible to reduce the message size. This parameter consists of three parts:

1) information about the size of the exponent,
2) the exponent, and
3) the modulus.

The RSA security is based on the size of the modulus. In the prototype solution, we have used the size of 128B (i.e. 1024b). Nowadays, such a size is not recommended by the NIST organization [18]. The minimal recommended size of the key is 2048b.

We have proposed two solutions, manual and automatic, each suitable for different kind of applications.

1) Manual solution: The manual solution is based on manual uploading of the mentioned HI-R information on the device. We can reduce the packet size by 128B from the whole size of the message (638B).

Advantages and disadvantages of the proposed solution:

- Re-generating and re-uploading of public parameters from the server to the device can be impossible. It depends on the quantity and distance of devices.
- Re-generating and re-uploading of public parameters is possible during the charging of some specific devices.
- It is necessary to regenerate keys on all devices at the same time.

The preconditions of the proposed solution:

- A simple access to all devices by the administrator (e.g. within a single building).
- It must be possible to power off the device for some time.

2) Automatic solution: For the automatic solution, an action of the administrator is not needed. The key is re-generated automatically in specified periods.

The basic working principle:

- Sending and uploading the parameters into a persistent memory of the device during the initial communication.
- Using the same parameters during the whole defined period. (Note: Even after restarting and re-initialization of the protocol).
- Re-generating, re-sending, and uploading of parameters after the specified period expire.

For both solutions, there are some possible applications. For example, the manual solution can be used within devices monitoring the patients' health during the clinical examinations, and the automatic solution can be used within devices monitoring the fullness of the container distributed around the city.

D. Removal of the HIT Parameter

The Host Identity Tag (HIT) is a 128-bit hash of the public key. Based on this hash value, the device can be identified using the same address-space size as in the IPv6 protocol. All the initialization messages of the HIP protocol contain two HIT fields, one for a responder and one for a sender. We propose to remove these fields and thus shorten the packet length. The following size of the messages would be reduced:

1) I1: 32B from 86B (i.e. 37%),
2) R1: 32B from 638B (i.e. 5%),
3) I2: 32B from 702B (i.e. 4.5%),
4) R2: 32B from 262B (i.e. 12.2%).

The HIT parameter is used in a variety of security functions, e.g. verifying the signature or puzzle mechanism, where it is used for validating the solution. The HIP puzzle mechanism protects the server from denial-of-service attacks [19]. To not decrease the security level of the protocol, we decided to not implement this solution in our prototype at this time. A similar solution is already implemented for data packets, where the HIT parameter is replaced by the SPI (Security Parameter Index) parameter.

E. Replacement of the SPI Parameter

The HIP protocol is used in combination with other protocols, such as IPSec.

The minimal requirement for data transmission in the protocol implementation is the use of the ESP transport mode. We also used this mode in the prototype. The ESP transport mode includes the SPI parameter for mapping the instance of the protocol and the corresponding SA (Security Association) to the relevant device, identified by the HIT parameter.

The proposed solution is to remove the ESP INFO parameters in the control messages and thus reduce the size of packets (i.e. 24B reduced in the I2 and R2 messages).

The identified disadvantages:

1) According to the HIP documentation, it is forbidden to map the instance to an IP address, which could happen if we implemented this proposal along with the previous one (i.e. the anonymous-location feature would be lost).
2) Limited protocol extensions enabling mapping of multiple SAs (identified by different SPI values) to the same device.

F. Optimization of Computing Tasks

The last proposed solution is an optimization of computing tasks, such as computing of Puzzle, hash, etc. We have identified two possibilities:

1) Optimization by Assembler: For better use of the processor and its resources, some parts of code (communication library, protocol stack) can be implemented in the assembler language. It is necessary a good knowledge of the IoT system and the assembler programming language. On the other side, nowadays, there are good compilers that make assembler-based optimization less effective.

2) Optimization by Hardware: This optimization means an optimization of computing tasks by dedicated hardware parts, supporting cryptographic, hash, and similar functionality. By using specialized hardware, the resource-intensive tasks can be processed more efficiently and the processor time can be spared. However, the energy efficiency of the hardware part must be carefully considered.

IV. Experimental Results

To verify the functionality of the proposed solution, we have implemented a prototype so far, as a proof-of-concept. The proposed modifications are implemented into an open-source implementation of the HIP protocol, called OpenHIP[1] [10]. Three of the six proposals were realized, namely Removal of the CloseAck message and the temporary closing state, Reduction of the parameter format and Removal of the HI-R parameter (the manual solution).

The prototype was tested on the topology illustrated in Fig. 2, which consists of two RPi 3 (Raspberry Pi) microcomputers (Raspbian operating system), interconnected by the

[1]Source code: https://github.com/rektide/openhip, version: openhip-0.9

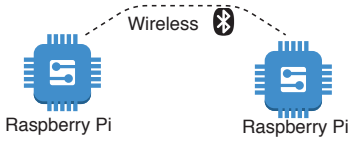

Fig. 2. The prototype architecture.

integrated Bluetooth 4.1 modules. The Bluetooth was used as part of the RPi and for the purpose of testing functionality.

After modifying the protocol by the above-mentioned optimizations (called E-HIP), the connection between two devices was successfully established and the communication worked correctly. Thus, the experiment has proved that the implemented optimizations are realizable and the protocol function is not corrupted. During the testing, we have measured the power consumption of the HIP protocol before (OpenHIP) and after (E-HIP) the modification. It represents an improvement of about 20%. The achieved results were compared to the existing works [17] in Fig. 3. Based on the comparison, we can tell that the achieved efficiency is comparable to other improvements. However, the proposed modifications are unique, and thus they can be further combined with other proposals to maximize the HIP energy efficiency.

V. Conclusions

There are some optimized security protocols used in IoT area, however, the battery-powered device lifetime is still limited. This was the target of our work. Based on the analysis, we decided to use the OpenHIP protocol and proposed its modification for efficient use in this area. The energy intensity of the proposed solution is comparable to other existing works. We can expect some contributions as reduction of network load, reduction of processor-time usage and reduction of energy required for communication. The modified protocol E-HIP is not compatible with standard one, but the application-specific communication requires the application-specific protocols. Also, there is no impact on the data plane, because the modification is focused only on the control plane.

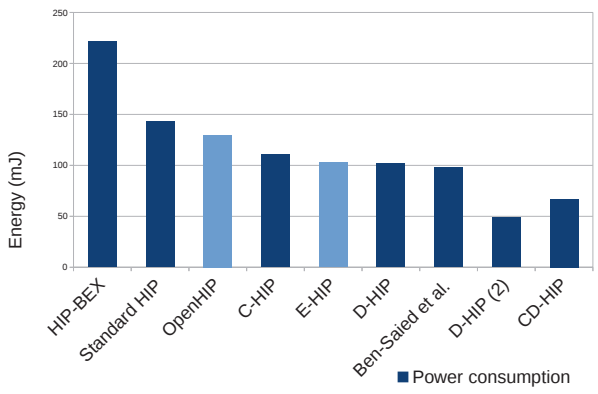

Fig. 3. The comparison of the power consumption.

In our further work, we are going to test it in a real IoT network using a low-power communication technology for IoT/mMTC (e.g. Bluetooth Low Energy, NB-IoT, LTE Cat-M1, etc.). The future work can also bring some possibilities for improvement since there are still unimplemented parts of the proposed solution that could be incorporated or it can be combined with some existing optimizations (e.g. C-HIP, D-HIP) and achieve even higher energy efficiency.

References

[1] H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelfflé, "Vision and challenges for realising the Internet of Things," *Cluster of European Research Projects on the Internet of Things, European Commision*, vol. 3, no. 3, pp. 34–36, 2010.

[2] J. Bartje, "The top 10 IoT application areas – based on real IoT projects," *IoT analytics*, 2016.

[3] B. Dorsemaine, J.-P. Gaulier, J.-P. Wary, N. Kheir, and P. Urien, "Internet of Things: A definition & taxonomy," in *2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies*. IEEE, 2015, pp. 72–77.

[4] O. El Mouaatamid, M. Lahmer, and M. Belkasmi, "Internet of Things security: Layered classification of attacks and possible countermeasures," *Electronic Journal of Information Technology*, no. 9, 2016.

[5] J. Morgan, "A simple explanation of 'The Internet of Things'," *Forbes*, May 2014.

[6] U. Banerjee, C. Juvekar, S. H. Fuller, and A. P. Chandrakasan, "eeDTLS: Energy-efficient datagram transport layer security for the Internet of Things," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 2014–2019.

[7] S. Raza, H. Shafagh, K. Hewage, R. Hummen, T. Voigt *et al.*, "Lithe: Lightweight secure CoAP for the Internet of Things," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3711–3720, 2013.

[8] M. Lavanya and V. Natarajan, "Lightweight key agreement protocol for IoT based on IKEv2," *Computers & Electrical Engineering*, vol. 64, pp. 580–594, 2017.

[9] R. Moskowitz, T. Heer, P. Jokela, and T. Henderson, "Host identity protocol version 2 (HIPv2)," Tech. Rep., 2015, rFC 7401. [Online]. Available: https://www.rfc-editor.org/info/rfc7401

[10] (2018) OpenHIP: Host identity protocol implementation. [Online]. Available: https://bitbucket.org/openhip/openhip

[11] H. Hellaoui, M. Koudil, and A. Bouabdallah, "Energy-efficient mechanisms in security of the internet of things: A survey," *Computer Networks*, vol. 127, pp. 173–189, 2017.

[12] P. Urien, "Innovative TLS/DTLS security modules for IoT applications: Concepts and experiments," in *International Internet of Things Summit*. Springer, 2015, pp. 3–15.

[13] R. Bonetto, N. Bui, V. Lakkundi, A. Olivereau, A. Serbanati, and M. Rossi, "Secure communication for smart IoT objects: Protocol stacks, use cases and practical examples," in *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2012, pp. 686–692.

[14] Y. B. Saied and A. Olivereau, "HIP Tiny Exchange (TEX): A distributed key exchange scheme for HIP-based Internet of Things," in *Third International Conference on Communications and Networking*. IEEE, 2012, pp. 1–8.

[15] R. Moskowitz and R. Hummen, "HIP Diet EXchange (DEX)," Tech. Rep., 2017, draft-ietf-hip-dex-06. [Online]. Available: https://tools.ietf.org/html/draft-ietf-hip-dex-06

[16] R. Hummen, J. Hiller, M. Henze, and K. Wehrle, "Slimfit-A HIP DEX compression layer for the IP-based Internet of Things," in *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2013, pp. 259–266.

[17] S. Sahraoui and A. Bilami, "Efficient HIP-based approach to ensure lightweight end-to-end security in the internet of things," *Computer Networks*, vol. 91, pp. 26–45, 2015.

[18] E. Barker, "Recommendation for key management part 1: General (revision 4)," *NIST Special Publication*, vol. 800, no. 57, pp. 1–160, 2016.

[19] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host identity protocol," Tech. Rep., 2008, rfc 5201. [Online]. Available: https://www.rfc-editor.org/info/rfc5201