

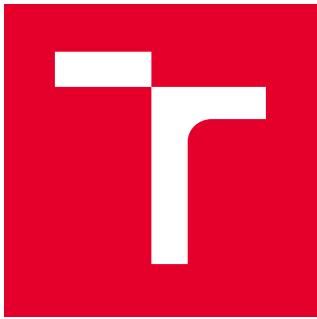
BRNO UNIVERSITY OF TECHNOLOGY

**Faculty of Electrical Engineering
and Communication**

BACHELOR'S THESIS

Brno, 2020

Jaromír Bača



BRNO UNIVERSITY OF TECHNOLOGY

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ

DEPARTMENT OF TELECOMMUNICATIONS

ÚSTAV TELEKOMUNIKACÍ

END-TO-END ENCRYPTION PROTOCOL FOR IEEE 802.15.4

PROTOKOL S KONCOVÝM ŠIFROVÁNÍM PRO IEEE 802.15.4

BACHELOR'S THESIS
BAKALÁŘSKÁ PRÁCE

AUTHOR
AUTOR PRÁCE

Jaromír Bača

SUPERVISOR
VEDOUCÍ PRÁCE

Ing. Ondřej Krajsa, Ph.D.

BRNO 2020

Bachelor's Thesis

Bachelor's study field **Information security**

Department of Telecommunications

Student: Jaromír Bača

ID: 133372

Year of study: 3

Academic year: 2019/20

TITLE OF THESIS:

End-to-end encryption protocol for IEEE 802.15.4

INSTRUCTION:

Design and implement a secure key exchange protocol for the AES128 encryption algorithm. This protocol will use the IEEE802.15.4 as data link protocol and the Atmel LighWeight Mesh as network protocol for communication. Implement the proposed protocol as well as the data link and network protocols on the AVR ATMega128RFA1 microcontroller. Measure the time required to exchange the key and the time required to encrypt and decrypt the information. As a part of this work, perform a security analysis of the proposed protocol and a comparison with similar techniques. Next, analyse and design a key exchange between the wireless node and the Internet element using the proposed protocol.

RECOMMENDED LITERATURE:

[1] LAVANYA, M. a V. NATARAJAN. Lightweight key agreement protocol for IoT based on IKEv2. Computers and Electrical Engineering [online]. Elsevier, 2017, 64 [cit. 2019-09-16]. DOI: 10.1016/j.compeleceng.2017.06.032. ISSN 0045-7906.

[2] THAMES, Lane a Dirk SCHAEFER. Cybersecurity for Industry 4. 0: Analysis for Design and Manufacturing. Cham: Springer, 2017. DOI: 10.1007/978-3-319-50660-9. ISBN 9783319506593.

Date of project specification: 3.2.2020

Deadline for submission: 8.6.2020

Supervisor: Ing. Ondřej Krajsa, Ph.D.

prof. Ing. Jiří Mišurec, CSc.
Subject Council chairman

WARNING:

The author of the Bachelor's Thesis claims that by creating this thesis he/she did not infringe the rights of third persons and the personal and/or property rights of third persons were not subjected to derogatory treatment. The author is fully aware of the legal consequences of an infringement of provisions as per Section 11 and following of Act No 121/2000 Coll. on copyright and rights related to copyright and on amendments to some other laws (the Copyright Act) in the wording of subsequent directives including the possible criminal consequences as resulting from provisions of Part 2, Chapter VI, Article 4 of Criminal Code 40/2009 Coll.

DECLARATION

I declare that I have written the semestral project titled "Protokol s koncovým šifrováním pro IEEE 802.15.4" independently, under the guidance of the advisor and using exclusively the technical references and other sources of information cited in the project and listed in the comprehensive bibliography at the end of the project.

As the author I furthermore declare that, with respect to the creation of this semestral project, I have not infringed any copyright or violated anyone's personal and/or ownership rights. In this context, I am fully aware of the consequences of breaking Regulation § 11 of the Copyright Act No. 121/2000 Coll. of the Czech Republic, as amended, and of any breach of rights related to intellectual property or introduced within amendments to relevant Acts such as the Intellectual Property Act or the Criminal Code, Act No. 40/2009 Coll., Section 2, Head VI, Part 4.

Brno
.....
author's signature

Contents

Introduction	5
1 Hardware	6
1.1 Mirocontrollers	6
1.2 deRFnode 1TNP2 DBT	7
1.3 Atmel Studio IDE	8
2 Standard 802.15.4	9
2.1 Topology	10
2.1.1 Star	10
2.1.2 Mesh (Peer-to-peer)	11
2.2 Atmel lightweight Mesh	11
3 IKEv2 - Internet Key Exchange	12
4 Elliptic-curve Diffie–Hellman	13
4.1 Theory of elliptic curve	14
5 Key exchange algorithm	17
5.1 Phase A	17
5.2 Phase B	19
5.3 Phase C	20
5.4 Metoda ověření pseudonáhodně vybraných vstupů	21
5.5 Výpočet prvního bodu a řádu grupy	21
5.6 Matematické funkce	22
Bibliography	25
List of acronyms	27
List of appendices	28

Introduction

The future belongs to automation. We have all come across such concepts as smart cities, self-driving cars, or fully-automated factories that do not need human beings to operate them. These things, which sounded like science fiction twenty years ago, are becoming a reality today. In addition, due to falling hardware prices, automation is not just a privilege for large corporations, but is becoming increasingly commonplace.

One of the pillars of automation is the collection and flow of data, which introduces a new concept - Internet of Things. Under this term we can imagine a large number of small devices that are often not even connected to the mains and are powered by batteries or solar panels. They use wireless networks to communicate with the environment, they use wireless networks that are specially designed for these small devices due to their limited power options.

Every new technology brings advantages and disadvantages. The main problem of wireless networks is their vulnerability to attacks. Input or output data can be both tapped and altered. This can result in a number of situations, i.e. from unpleasant to fatal. The logical response will be to use some kind of network security. But given the fact that most small devices are dedicated and built on microcontrollers, it is necessary to use an adequate lightweight solution.

Structure of the semestral thesis

The Semestral Thesis is divided into a theoretical part, a practical part and a conclusion. In the theoretical part the AVR microcontrollers will be described in depth. We will talk about their history and how to program them. Next, a lightweight mesh network suitable for small, low-energy devices will be described. The theoretical part concludes with a chapter devoted to the description of ELDH, SHA 1 and AES used. In the practical part the created protocol will be implemented and tested on the deRFnode 1TNP2 DBT development platform from the German manufacturer Dresden Elektronik Verkehrstechnik GmbH. The conclusion of the thesis is devoted to the evaluation of achieved results.

1 Hardware

This Semestral Thesis is practically oriented and uses a number of special software and hardware instruments. In this chapter, the microcontrollers, which form the core of the used deRFnode 1TNP2 DBT boards, will be described in detail. The conclusion of the chapter deals with AtmelStudio 7 Development Environment.

1.1 Mirocontrollers

In today's world, we are surrounded by various small smart devices that can record, for example, ambient temperature, or simple machines that perform one or a limited number of defined activities. All these devices work thanks to the small built-in mini computers we call microcontrollors, or MCU¹ for short.

Although microcontrollers look like processors known from PC assemblies, they are fully functional computers. In the case of computational operations, no other peripheral input and output devices can be used. It is sufficient to provide them with electricity. In addition to the CPU itself, they include RAM and EEPROM memory to store the code that the computer executes. In this Semestral Thesis, the ATmega128RFR2 chip is used, which is directly embedded in the deRFnode 1TNP2 DBT development board.

They are equipped with serial ports for basic communication with peripherals, which may be other microcontrollers or other electronic devices, such as sensors, servomotors. Selected ports are grouped into interfaces, such as SPI - Peripheral Interface, which allows full duplex data communication between two microcontrollers, where on one side there is a master that controls the slave microcontroller on the other. This interface can be used for computer-to-microcontroller communication, but it requires a special converter. Another well-known interface is the I²C, also known as TWI², which allows two devices to be connected in series with two wires at a time and to communicate using address data.

Among the world's leading manufacturers of microcontrollers are companies such as Texas Instruments, Microchip Company (former Atmel), Interl Corp., Fujistru, and others.

¹MicroController Unit

²Two Wire Interface

1.2 deRFnode 1TNP2 DBT

It is a development board that includes a radio module and an ATmega 128 chip. This board is designed for low-energy data networks such as 802.15.4. The board has a number of interfaces such as USB, JTAG or TWI which. In addition to powering via a USB cable connected to the computer, the board can be powered with a 5V DC plug. The board is also equipped with a battery pack for three AAA batteries, which allow the board to operate without the need to connect to the power grid.

The board is produced in two variants, deRFgateway and deRFnode. The first is equipped with an Ethernet interface. This board can be used as a network co-ordinator that collects data from other nodes and sends it to another network, in this case to an 802.3 Ethernet network. The second type deRFnode can serve as a coordinator within the WSN network or as an RFD node. The board contains sensors that measure celeration, temperature and luminosity.

The main advantage of these boards is their variability. The radio module is removable from the board and can be replaced with another compatible type from Dresden Elektronik Verkehrstechnik, GmbH. Another optional part is used software, where we can choose between different network stacks from the manufacturer or external developers.

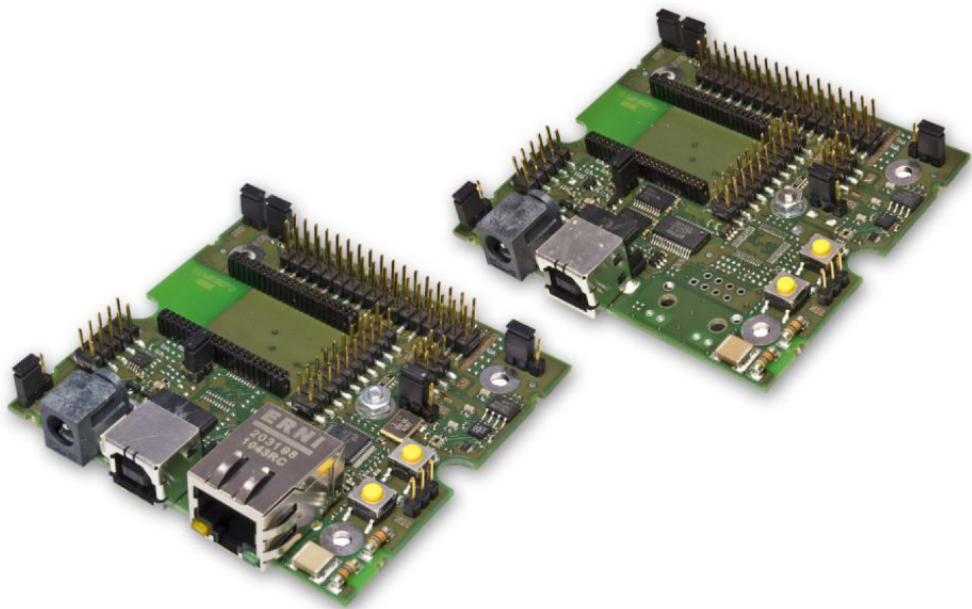


Fig. 1.1: deRFgateway and deRFnode board (without radio modules) [2]

1.3 Atmel Studio IDE

The software part of the term paper was realized in the development environment of Atmel Studio 7 IDE³. This development environment is intended for development and debugging of applications written in C/C++ languages. Studio allows to program any of over 500 supported AVR and SAM microcontrollers via USB programmers, for example Atmel ICE.

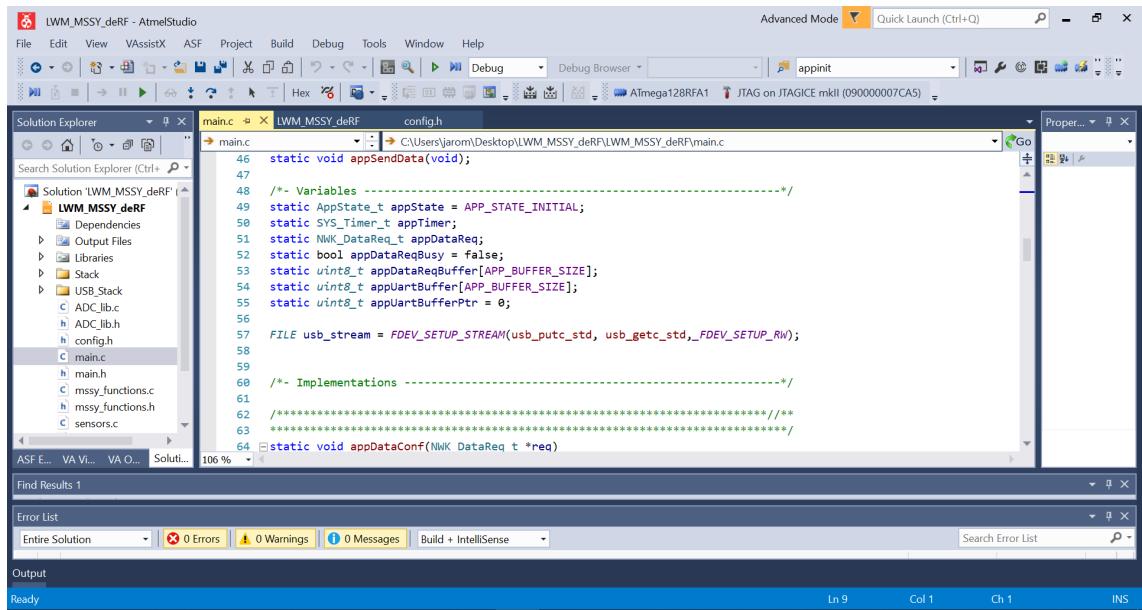


Fig. 1.2: Atmel Studio 7 IDE

³IDE - integrated development environment

2 Standard 802.15.4

The main motivation for the design of IEEE 802.15.4 was to create a communication standard for WPAN networks that would be optimized for low-energy devices for use in industrial automation. This standard serves as a basis for higher protocols, such as ZigBee, WirelessHard, 6LoWPAN, etc. The OSI model defines the link and physical layer parameters. Higher layer protocols are not specified.

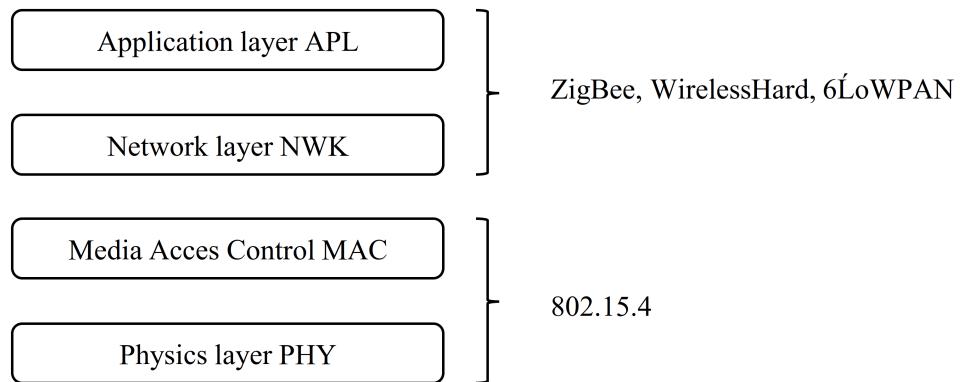


Fig. 2.1: Lyers of 802.15.4 and higher protocols

The physical layer

The general task of the physical layer is to transmit data. This layer defines the frequency band and modulation used. Within the physical layer we distinguish a total of three frequency bands with different transmission speed and number of channels.

- Europe 868.0 – 868.8 MHz – pouze jeden kanál (0), 20 – 250kbit/s
- North America 902 – 928 MHz - 13 channels (1-14)
- Worldwide 2400–2483.5 MHz - 16 channels

Data link layer

The link layer ensures correct addressing of forwarded data. Other tasks include, for example, synchronization according to the beacon frame, which it transmits at regular intervals and thus lets you know about the presence of the network.

2.1 Topology

The standard uses star and mesh topologies. Topologies consist of two basic types of devices, FFD Fully Function Device (RFD) and RFD (Reduced FunctionDevice).

FFD (Fully Function device)

This device can serve either as a network coordinator, a terminal coordinator, or as a terminal only. In the case of the first role, the device acts as a router and in addition to network management can forward data to other networks based on other standards, such as ethernet, wifi.

RFD (Reduced Function device)

An RFD device is a feature with reduced functionality and only works as an endpoint that only receives or sends data to its coordinator, not to another point on the network.

2.1.1 Star

This type of topology consists of one network coordinator to which devices that can be FFD or RFD are connected but only communicate with the network coordinator

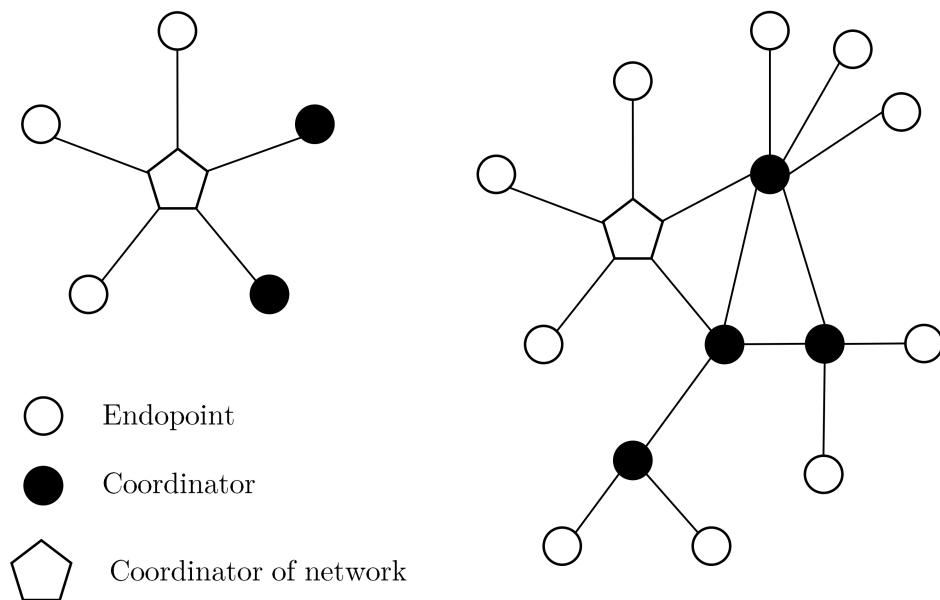


Fig. 2.2: Topologies Star and Mesh (Peer-to-peer) [1]

2.1.2 Mesh (Peer-to-peer)

If there is no requirement for peer-to-peer topology to send data to other networks, there is no need to include a network coordinator in the network. The advantage of this solution is the possibility to build a network that can have a higher range than the network coordinator radio module in the case of a star topology. This is made possible by the ad hoc capability where the data packet is forwarded from the sender to the recipient through several intermediate nodes. However, this solution has a negative effect on the energy consumption of the system.

2.2 Atmel lightweight Mesh

To use our protocol design, Atmel's Lightweight Mesh SDK was used for use in a low-power wireless network. You can apply it to any system or development board that works with an MCU with the hLow Power transceiver for 802.15.4, such as the ATmega128RFA1 that is used in the deRFnode 1TNP2 DBT board. It is possible that, based on this protocol, it will theoretically have up to 65635 nodes [1].

3 IKEv2 - Internet Key Exchange

The purpose of this protocol is to secure communication between the two parties, not only by securing the forwarded data, but also by securing the communication channel. Communication is thus secured against data theft or usage of fraudulent data.

The principles of the IKEv2 is depicted in Figure 3.1. Basically, the protocol is divided into three main parts. In the first part there will be mutual exchange of keys based on the Diffie-Hellman algorithm. Side A will propose to side B various combinations of SA - Security Association, which are a set of algorithms used to encrypt and authenticate the subscribers. The party B will choose the most appropriate combination of SA based on its capabilities. In the second phase of the protocol, authentication of the parties, key exchange and activation of the agreed encryption algorithm according to the selected SA will take place. In the third phase, when both parties receive an identification tag (SPI), which confirms the identity of the forwarded data. The key exchange will then be used again, which will be used to encrypt the transmitted data.

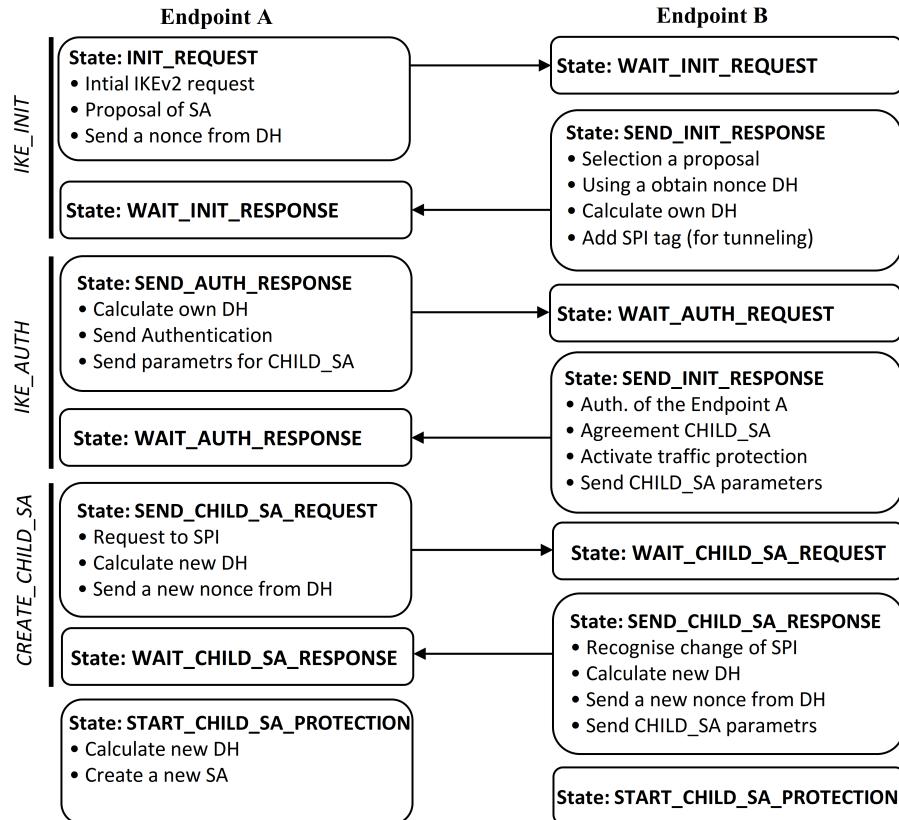


Fig. 3.1: IKEv2 key exchange [5]

4 Elliptic-curve Diffie–Hellman

In the previous chapter was described the methodology of key exchange. The key exchange itself is realized by asymmetric cryptography, where both parties independently determine the secret key from which they calculate the public key by means of a one-way mathematical function. With a one-way function, it is easy to calculate the public key from the secret key, but to recalculate the secret key from the public key is mathematically very difficult. This method was discovered by Whitfield Diffie and Martin Hellman in the 1970s. Although it later turned out that the method had been invented a few years earlier by the British intelligence and security organization GCHQ, that kept this fact secret until the 1990s. That is why this key exchange protocol is known as the Diffie-Hellman protocol.

Tab. 4.1: Key size comparison between Diffe-Hellman algorithm and ECHD [5]

Key Size in bits (by NIST recommendation)	Diffie-Hellman algorithm (modulus size on bits)	ECC size
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

This protocol can be used with different key lengths, as described in the table 4.1, where, among other things, a key size equivalent is added for a better overview, as recommended by the NIST¹. However, the size of keys of the classic Diffie-Hellman protocol is unsuitable for application on low-power devices because of the transmission of larger amounts of data, which has a negative effect on power consumption and the need for more storage space. To solve this problem an elliptic curve-based algorithm can be used. Their main advantage is a smaller key size when compared to the classic Diffie-Hellman algorithm, while maintaining the same level of security. For example, a 224-bit key created by an elliptical curve-based algorithm provides equivalent security as a 2048-bit key generated by the Diffie-Hellman algorithm.

¹National Institute of Standards and Technology

4.1 Theory of elliptic curve

In this part of the thesis the theory of elliptic curves is briefly described. Although there are more theories and computational methods, we will confine ourselves to describing an elliptic curve on the $\text{GF}(p)$ type of the division ring that uses modular arithmetic. The theory described here is based on publications [8]. This theory has been used as a basis for creating a key exchange algorithm in this term paper.

Elliptic curve is plane algebraic curve defined by an equation 4.1, where the values X and Y represent the Cartesian coordinates of the chosen starting point, and a and b are curve parameters.

$$y^2 = x^3 + ax + b \quad (4.1)$$

However, before calculating the other points on the curve, it is necessary to verify that the proposed point actually lies on the curve. This can be easily verified using a modified form of equation 4.2, where modular arithmetic is used. If the right and left calculations are the same, then it is sure that the point lies on the curve. The value p is and prime. This verification was carried out within the framework of this term paper and was implemented by software and it is included in the appendices.

$$(y^2) \bmod p = (x^3 + ax + b) \bmod p \quad (4.2)$$

The points on the elliptical curve are made up of an additive group, where each additional point arises by adding up the previous point and the starting point. It means logically that there are two different methods to sum the two points accordingly whether the added points are the same or different.

Addition of two identical points

This method, also known as doubling, is usually used to calculate the second point in a sequence within a group, where we add the starting point with itself. Graphical solution of this method is shown in the picture 4.1.

$$S = \frac{3x_P^2 + a}{y_P} \bmod p \quad (4.3)$$

$$x_R = s^2 - 2x_P \bmod p \quad (4.4)$$

$$y_R = s(x_P - x_R) - y_P \bmod p \quad (4.5)$$

Using these formulas we are able to calculate a common point. By calculating the first equation we obtain the slope of the curve S . We then use this value to calculate the coordinates x_R and y_R .

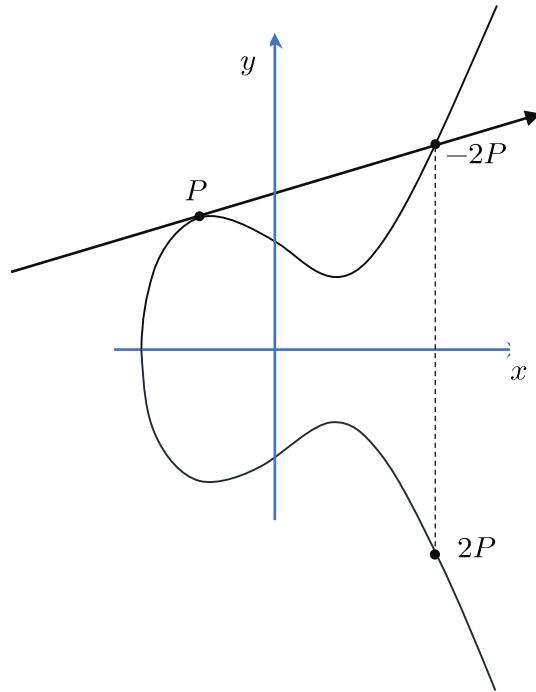


Fig. 4.1: Doubling [8]

Addition of two different points

As in the previous case we must first calculate the slope of the curve S and then coordinates x_R and y_R . The figure below shows again the graphical method of finding a common point

$$S = \frac{y_P - y_Q}{x_P - x_Q} \quad (4.6)$$

$$x_R = s^2 - (x_P + x_Q) \quad (4.7)$$

$$y_R = s(x_P - x_R) - y_P \quad (4.8)$$

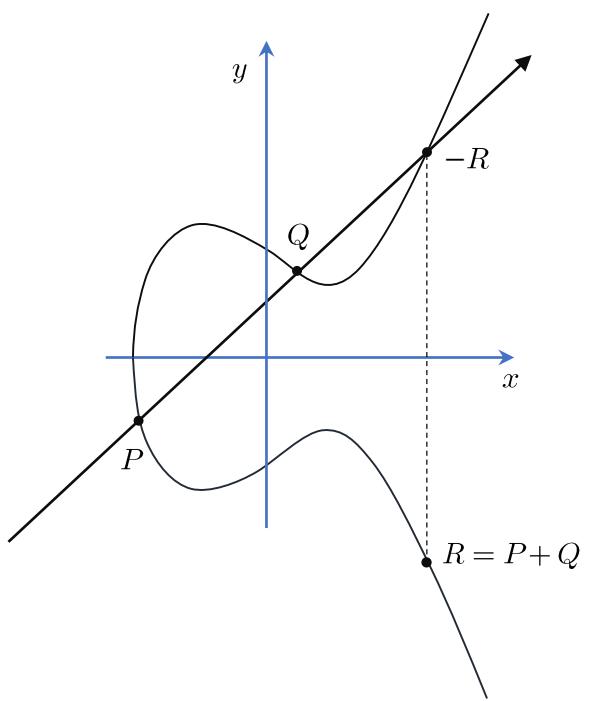


Fig. 4.2: Addition point [8]

5 Key exchange algorithm

V této kapitole si postupně popíšeme fungování algoritmu pro výměnu klíčů. Algoritmus je založen na eliptických křivkách a na Weierstrass metodě výpočtu eliptických křivek, která je popsána v kapitole (—) a je navržen pro práci v dekadické soustavě. Výměna klíčů je rozdělena do tří fází, které je ilustrováno na obrázku.. První fáze je spuštěna iniciátorem komunikace, kde dojde k pseudonáhodnému výběru vstupních hodnot, které se odešlou ve formě prostého textu k příjemci. Ve druhé fázi, která již probíhá u obou komunikačních uzlů, dojde k výběru tajného klíče a výpočtu bodu, který si každá strana vymění s protistranou. Z obdrženého bodu od protistrany je každý z účastníků již schopen získat společný klíč. Tento proces je sprostředkován třetí fází. Komunikace mezi jednotlivými fázemi a další využití klíčů je řešeno v další kapitole, která pojednává o implementaci aplikace pro výměnu klíče do lightweight stacku.

V další fázi výměny klíčů jsou ověřené vstupní hodnoty odeslány ve formě zašifrovaného payloadu uzlu B. Je výhodnější posílat ověřené vstupní hodnoty, z důvodu menšího počtu potřebných dat k odeslání. Tím se šetří energie nutná pro odeslání dat na fyzické vrstvě. Uzel B ze znalosti těchto hodnot sám vypočítá první bod a řád grupy. Tentokrát již není kontrola výpočtu zapotřebí. Po provedení této fáze, jsou oba komunikační uzly schopné přistoupit na výměnu klíče.

5.1 Phase A

Tato fáze je pracuje pouze na straně inciátora komunikace. Jejím úkolem je výběr vhodných hodnot, které budou použitelné pro výpočet grupy bodů na eliptické křivce. Výběr hodnot je zcela nezávislý na jakékoliv tabulce s předchystanými hodnotami. První části této fáze algoritmu dochází k výběru čísla, které představuje hodnotu modulo. Dle teorie je dáno, že toto číslo musí být prvočíslo. Proto je po výběru čísla do algoritmu zařazen blok, který testuje číslo, zda je prvočíslem či nikoliv. Pokud zjistí, že číslo není prvočíslem, dojde k opakování smyčky, tedy opětovnému výběru a testu. Jeli výběr úspěšný, následuje výběr parametrů a a b, které představují asymptoty eliptické křivky. Vybrané hodnoty se testují dle rovnice. V případě negativního výsledku testu, dochází k opakování smyčky dokud nejsou vybrány vhodné hodnoty. V další části této fáze aplikace jsou vypočteny souřadnice prvního bodu, který představuje generátor grupy. Tento bod je poté podroben testu, kdy je ve smyčce znova vypočítána celá grada, tentokrát ze znalosti generátoru. Ověřuje se, že body lze sečíst a výsledek není bod v nekonečnu. Pokud je testem nalezena chyba, aplikace se ve smyčce vrací na začátek, kde od znova

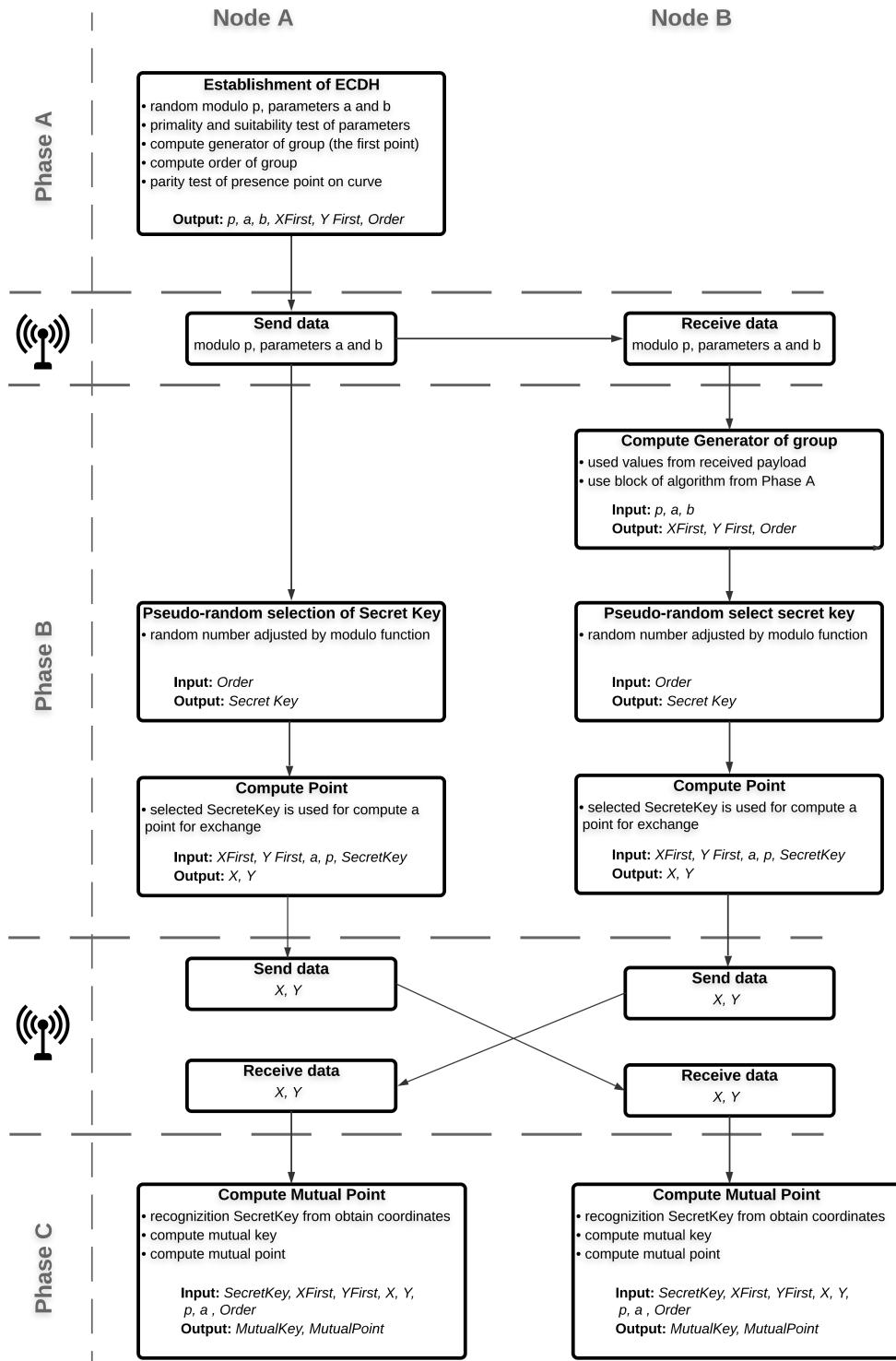


Fig. 5.1: Key exchange algorithm

proběhne výběr prvočísla a parametrů eliptické křivky. Pokud však je test úspěšný následuje další test, kdy je vypočítaný poslední bod testován, zda koordináty X a Y skutečně leží na křivce. Toto řešení bylo zvoleno z důvodu problémů během tvorby algoritmu, kdy i přes vhodně zvolené a otestované vstupní hodnoty byly vypočítánné grupy bodů neparitní a tudíž pro asymetrickou výměnu klíčů nepoužitelné. Výstupem této fáze je ověřené prvočíslo, asymptoty křivky, souřadnice prvního bodu a řád grupy.

Method of obtain point					
p modulo a, b parametry	x	$x^3 + ax + b$	y	y^2	$[X, Y], [X, Y]$
	0	2	2	6	[0, 6], [0, 11]
	1	5	-	-	-
	2	14	-	-	-
Example, we have:	3	1	1	1	[3, 1], [3, 16]
$p = 17$	4	6	-	-	-
$a = 2$	5	1	1	1	[5, 1], [5, 16]
$b = 2$	6	9	9	3	[6, 3], [6, 14]
	7	2	2	6	[7, 6], [7, 11] $\Rightarrow (7^3 + 2 \cdot 7 + 2) \bmod 17 = 2$
	8	3	-	-	-
	9	1	1	1	[9, 1], [9, 16]
	10	2	2	6	[10, 6], [10, 11] $2 \Rightarrow 6 \Rightarrow [7, 6], [7, 11]$
	11	12	-	-	-
	12	3	-	-	-
	13	15	15	7	[13, 7], [13, 10]
	14	3	-	-	-
	15	7	-	-	-
	16	16	16	4	[16, 4], [16, 13]
	±5	8	-	-	0
	±6	2	-	-	-
	±7	15	-	-	-
	±8	13	-	-	-
	$\Rightarrow 6^2 \bmod 17 = 2$				
	Order of group is 19 (18 points + zero point)				

Fig. 5.2: Method of computation [11]

5.2 Phase B

Tato fáze má za úkol výběr klíče a výpočet bodu, který představuje veřejný klíč a bude odeslán protistraně. Aplikace se dělí na dvě subverze s ohledem, kde je používána. Pokud je použita na straně iniciátora komunikace, obsahuje pouze blok pro generování pseudonáhodného tajného klíče a výpočet bodu pro sdílení. V případě subverze pro příjemce komunikace, je do aplikace implementován blok, který ze znalosti prvočísla a asymptot křivky vypočítá první bod a řád grupy. Tento výpočet není nijak ověřován. Je zde důvěra, že obdržené hodnoty od iniciátora jsou použitelné pro výpočet. V případě, že by tyto hodnoty byly podvržené, nemůže proběhnout úspěšná výměna klíčů.

5.3 Phase C

Tato závěrečná fáze algoritmu následuje po vzájemné výměně bodů, které představují veřejné klíče. První blok této aplikace umí z obdrženého klíče zpětně rozeznat jeho pořadí v grupě a tím získat tajný klíč protistrany. Druhá fáze má za úkol svůj vlastní tajný klíč a tajný klíč, který se získal z obdrženého bodu, vynásobit a získat tak společný klíč. Následně je z tohoto klíče spočítán společný bod, který se použije jako základ klíče pro šifrovací algoritmus AES 128.

5.4 Metoda ověření pseudonáhodně vybraných vstupů

Na záčatku Phase A jsou provedeny trsty pseudonáhodných čísel...

5.5 Výpočet prvního bodu a řádu grupy

Použití této části algoritmu je podmíněno úspěšným průběhem předchozí části algoritmu, který vygeneroval zaručeně funkční grupu bodů a je tedy znám první bod grupy a jeho řád. Výpočet bodu, který je součtem dvou bodů probíhá následujícím způsobem. V úvodní části výpočtu algoritmus na základě hodnot rozhodne, zda se jedná o součet dvou stejných bodů nebo dvou rozdílných bodů, popřípadě zda součet dle teorie eliptických křivek jedná o nulový bod. Z metody výpočtu je zřejmé, že první součet je součet první a prvního bodu. Tedy se jedná o součet stejného bodu. V další iteraci proběhne součet výsledku a počátečním bodem. Díky návrhu není nutné složitě sestavovat posloupnost použití jednotlivých metod, která v některých případech může vést k chybným výsledkům. Vhodná metoda výpočtu je situaci volena samotným programem. V této části aplikace jsou obsaženy části kódu, které představují programovou implementaci vzorců pro výpočet lambda a souřadnic X a Y. V průběhu výpočtu jsou dle potřeby volány matematické funkce, které jsou popsány v podkapitole (==).

Je nutné zmínit, že do aplikace vstupuje i proměnná, která sebou nese velikost grupy. Prakticky se jedná o počet opakování, které by vedly k výpočtu všech bodů grupy. Pokud však číslo bude menší, výsledkem bude jiný bod z grupy. Tímto způsobem na základě náhodně zvoleného čísla, který představuje tajný klíč. Vypočtený bod bude naopak představovat veřejný klíč a odešle se protistraně.

5.6 Matematické funkce

V průběhu výpočtu je často zapotřebí použít matematických funkcí jako je mocnění a modulární aritmetika. Tyto funkce jsou obsaženy v jazyce C. Ačkoliv tyto funkce komplikátor bez námitek zkompiluje, vlastnosti těchto funkcí nejsou vhodné pro použití v našem algoritmu a jejich použití by mohlo vést k nesprávným výsledkům. Pro zajištění spolehlivosti byly pro nadefinovány následující funkce

Funkce pow, která má za úkol mocnění čísel, pracuje s datovým typem double. V našem algoritmu pracujeme pouze s celými čísly. Pokud použijeme na vstupu do funkce pow celá čísla, může to mít v některých situacích nežádoucí výsledek, na příklad druhá mocnina čísla 5 je 24 namísto správného výsledku 25. Pro předejítí takovým problémů byla vytvořena aplikace pro mocnění, která pracuje s celými čísly a je schopna mocnit i záporná čísla. Aplikace je vytvořena jako samostatný modul, který je v procesu výměny klíčů hlavním programem dle potřeby volán.

Další matematickou aplikací je algoritmus pro výpočet zbytku pomocí modulární aritmetiky. Tato funkce je obdobně jako mocnění součástí jazyka C. Ovšem tato funkce nepracuje se zápornými čísly, které se v průběhu výpočtů v našem hlavním algoritmu vyskytují. Proto byla vytvořena aplikace, která po zavolání a obdržení vstupů provede výpočet a vrátí zpět výslednou hodnotu. Vedle potřeby klasického výpočtu zbytku po dělení je tu i požadavek na inverzní variantu. V aplikaci je pomocí cyklu while prováděn výpočet tak dlouho, než se zbytek rovná 1. Počet opakování je zároveň výsledkem. Tato aplikace je opět navržena tak, že bez problému funguje i se zápornými čísly. Poslední z matematických aplikací je test prvočíselnosti. Při inicializaci hlavního algoritmu dochází k volbě hodnoty modulo pomocí pseudonáhodné funkce. Pseudonáhodný výběr generuje jakékoli reálné číslo. Ovšem pro náš výpočet je nutné, aby hodnota modulo byla prvočíslem.

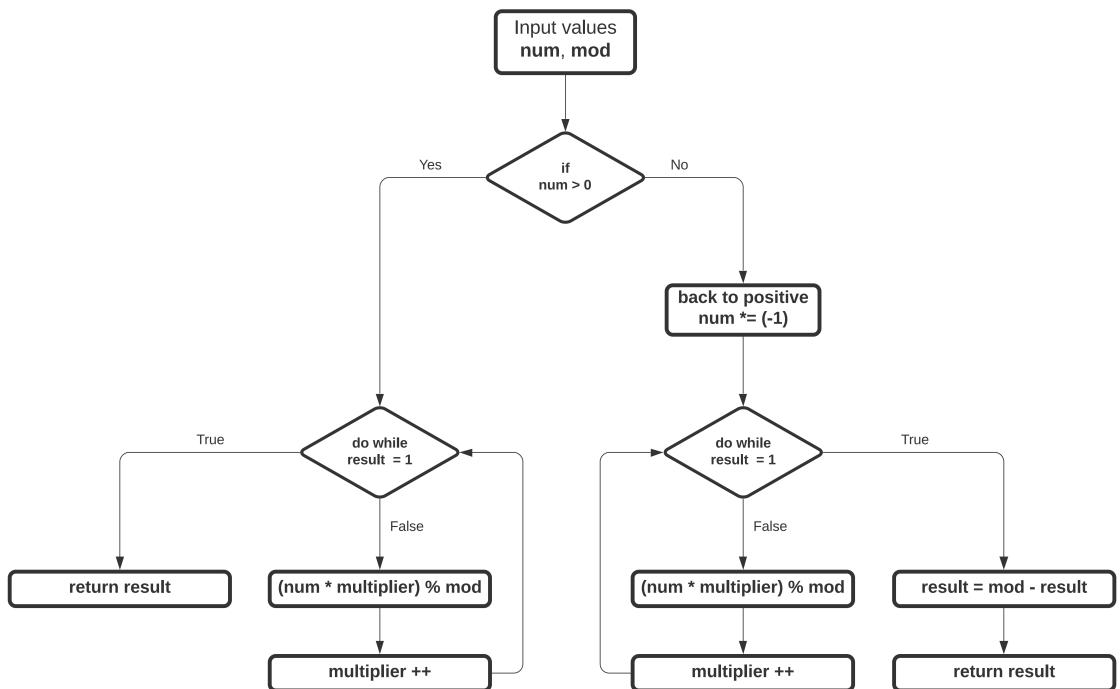


Fig. 5.3: Inverse modulo function [11]

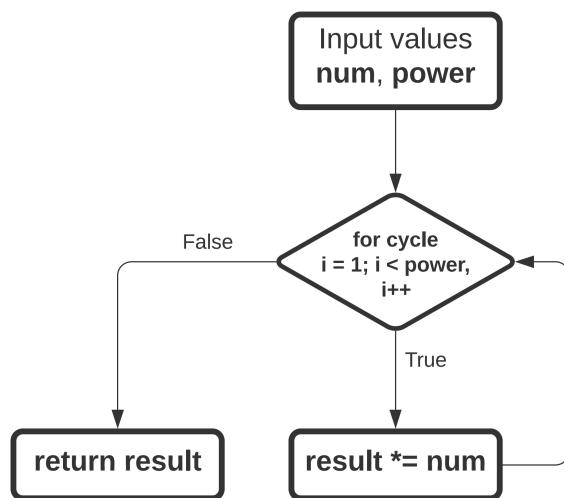


Fig. 5.4: Power function [11]

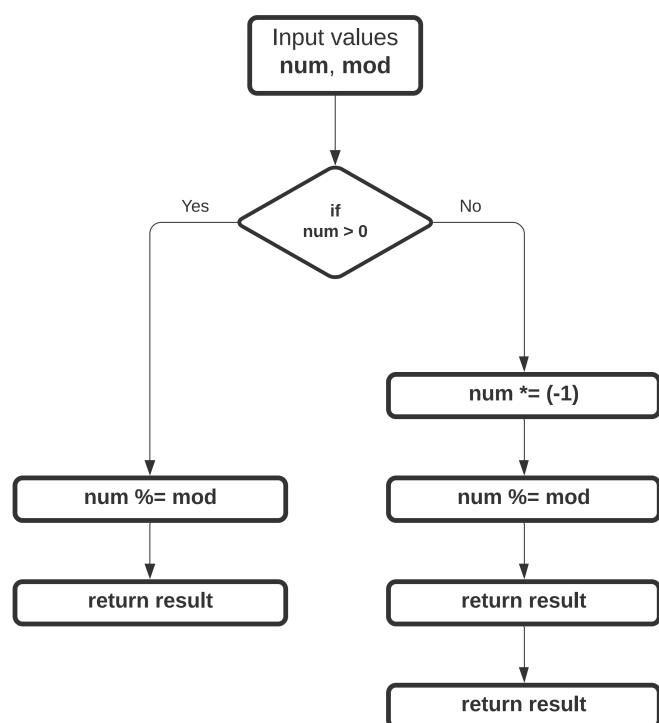


Fig. 5.5: Modulo funcktion [11]

Bibliography

- [1] *ATmega256RFR2 ATmega128RFR2 ATmega64RFR2 Datasheet*. [online], 2014. In: . Atmel (now Microchip Corporation), s. 611 [cit. 2019-12-02]. Available from: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8393-MCU_Wireless-ATmega256RFR2-ATmega128RFR2-ATmega64RFR2_Datasheet.pdf
- [2] *User manual - deRFnode / deRFgateway*, 2014. Dresden Elektronik, 56 s. Available from: https://www.dresden-elektronik.de/funktechnik/uploads/media/deRFnode_deRFgateway-BHB-en_10.pdf
- [3] MANN, Burkhard, 2003. *C pro mikrokontroléry: ANSI-C, komplátory C, spojovací programy - linkery, práce s ATMEL AVR a MSC-51, příklady programování v jazyce C, nástroje pro programování, tipy a triky ...* Praha: BEN - technická literatura. ISBN 80-730-0077-6.
- [4] MATOUŠEK, David, 2006. *Práce s mikrokontroléry ATMEL AT89C2051: [měření, řízení a regulace pomocí několika jednoduchých přípravků]*. Práce s mikrokontroléry ATMEL AVR ATmega16. Praha: BEN - technická literatura. ISBN 80-730-0048-2.
- [5] LAVANYA, M. and V. NATARAJAN, 2017. *Lightweight key agreement protocol for IoT based on IKEv2*. **64**, 580-594. DOI: 10.1016/j.compeleceng.2017.06.032. ISSN 00457906. Available from: <https://linkinghub.elsevier.com/retrieve/pii/S0045790617319286>
- [6] HEROUT, Pavel, 2011. *Učebnice jazyka C*. Dotlač 6. vyd. České Budějovice: Kopp nakladatelství. ISBN 978-80-7232-383-8.
- [7] LEVICKÝ, Dušan, 2016. *Kryptografia a bezpečnosť komunikačných sietí*. Košice: Elfa. ISBN 978-80-8086-254-1.
- [8] BURDA, Karel, 2015. *Úvod do kryptografie*. Brno: Akademické nakladatelství CERM, 66 s. ISBN 978-80-7204-925-7.
- [9] *Cybersecurity for industry 4.0*, 2017. New York, NY: Springer Berlin Heidelberg. ISBN 978-331-9506-593.
- [10] *C Program for Extended Euclidean algorithms* [online], In: . [cit. 2019-12-19]. Available from: <https://www.geeksforgeeks.org/c-program-for-basic-and-extended-euclidean-algorithms-2/>

- [11] *Secure Hash Algorithm (SHA-1)* [online], In: . [cit. 2019-12-19]. Available from: <http://www.hoozi.com/post/b3mf9/secure-hash-algorithm-sha-1-reference-implementation-in-c-c>

List of acronyms

AES	Advanced Encrypt S
JTAG	Joint Test Action Group
AVR	Alf and Vegard's RISC processor
SHA	Secure Hash Algorithm
IEEE	Institute of Electrical and Electronics Engineers
ISP	In System Programming
WSN	Wireless Sensor Network
IoT	Interner of Things
MCU	Interner of Things
EEPROM	Electrically Erasable Programmable Read-Only Memory
ESP	Encapsulating security protocol
NHC	Next header protocol
AH	Authentication header
LKA	LightWeight key agreement
ISAKMP	Internet Security Association and Key Management Protocol

List of appendices

ECDHverification.c

PointPP.c

PointPQ.c

Group of Points.c

ECDHduplex.c - (combination of algorithm PointPP. and PointPQ.c)

Protocol Stack - in file

IKE_MakeOfDecison.c

IKE_MutualPoint.c

FinalKeyExchangeAlgorithm.c