BIZTONSÁGOS KOMMUNIKÁCIÓS PROTOKOLLOK

EGYETEMI DOKTORI (PHD) ÉRTEKEZÉS

CSERNUSNÉ ÁDÁMKÓ ÉVA

TÉMAVEZETŐ DR. PETHŐ ATTILA

DEBRECENI EGYETEM
TERMÉSZETTUDOMÁNYI ÉS INFORMATIKAI DOKTORI TANÁCS
INFORMATIKAI TUDOMÁNYOK DOKTORI ISKOLA

DEBRECEN, 2020

SECURE COMMUNICATION PROTOCOLS

PH.D. THESIS

ÉVA CS. ÁDÁMKÓ

SUPERVISOR PROF. ATTILA PETHŐ

UNIVERSITY OF DEBRECEN
DOCTORAL COUNCIL OF NATURAL SCIENCES AND INFORMATION TECHNOLOGY
DOCTORAL SCHOOL OF INFORMATICS

DEBRECEN, 2020

Ezen értekezést a Debreceni Egyetem Természettudományi és Informatikai Doktori Tanács Informatikai Tudományok Doktori Iskola Elméleti számítástudomány, adatvédelem és kriptográfia programja keretében készítettem a Debreceni Egyetem természettudományi doktori (PhD) fokozatának elnyerése céljából.
Nyilatkozom arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.

Debrecen, 2020. június 18.

……..

Csernusné Ádámkó Éva
jelölt

Tanúsítom, hogy Csernusné Ádámkó Éva doktorjelölt 2010.- 2020 között a fent megnevezett Doktori Iskola Elméleti számítástudomány, adatvédelem és kriptográfia programjának keretében irányításommal végezte munkáját. Az értekezésben foglalt eredményekhez a jelölt önálló alkotó tevékenységével meghatározóan hozzájárult. Nyilatkozom továbbá arról, hogy a tézisekben leírt eredmények nem képezik más PhD disszertáció részét.

Az értekezés elfogadását javasolom.

Debrecen, 2020. június 18.

……..

Dr. Pethő Attila
témavezető

SECURE COMMUNICATION PROTOCOLS

Értekezés a doktori (Ph.D.) fokozat megszerzése érdekében
az informatika tudományágban.

Írta: Csernusné Ádámkó Éva okleveles matematika és informatika szakos tanár.

Készült a Debreceni Egyetem Informatikai Tudományok Doktori Iskolája
(Elméleti számítástudomány, adatvédelem és kriptográfia programja) keretében

Témavezető: Dr. Pethő Attila

A doktori szigorlati bizottság:

elnök: Dr. …… ……

tagok: Dr. …… ……

Dr. …… ……

A doktori szigorlat időpontja: 20…… …… …

Az értekezés bírálói:

Dr. …… ……

Dr. …… ……

Dr. …… ……

A bírálóbizottság:

elnök: Dr. …… ……

tagok: Dr. …… ……

Dr. …… ……

Dr. …… ……

Dr. …… ……

Az értekezés védésének időpontja: 20…… …… …

CONTENTS

1 INTRODUCTION

The final touches of the present dissertation were added during the COVID-19 pandemic in 2020. Governments all around the world have made significant efforts to stop the spread of the virus, usually by quarantining the infected people and by "social distancing". Both previous actions proved to be effective in flattening the Coronavirus Curve a.k.a. keeping the number of infected people below a critical value at a given time. In many countries, the violation of the restrictions has serious consequences, e.g. it can be resulted in an infringement procedure against the citizen. Thus, the valid location of the infected person has to be provided and proven unambiguously. For example, in South Korea, a unique tracking device is strapped to the wrist of the infected citizens to monitor their movement during the quarantine [1]. In Hungary, a mobile application is under development which is suitable for providing the information on whether a person was dangerously close to another infected one [2]. Both abovementioned methods are based on a GNSS system, such as the American GPS, or the Korean Positioning System (which will be launched soon) or the European GALILEO. However, the location information provided by the unique bracelet or the mobile device in the previous examples cannot be considered trusted from a cryptographical point of view. The question arises on how the location information can be authenticated in a cryptographic sense.

The first aim of the dissertation is to reveal the security problems of the GNSS systems, especially the GPS, and then overviewing the existing solutions for the revealed security breaches by comprehensively examining the related scientific literature. Additionally, to give solutions for the revealed security breaches and provide a way to authenticate location information from a cryptographical point of view.

During such critical periods – such as the outbreak caused by the virus COVID-19 – the proper and reliable operation of the critical infrastructures is crucial. Critical infrastructures such as the newly invented multi-user ventilators – used for respiratory support – controlled by PLC-s (as small SCADA system), or the public services, as emergency services, energy or water providers, or communication services (as large SCADA systems) are indispensable during "peacetime" also.

The second aim of the dissertation is to uncover the vulnerabilities of one of the basic building blocks of such SCADA systems mentioned above, namely the Modbus RTU industrial communication protocol, and studying the existing solutions for the revealed vulnerabilities by comprehensively examining the related scientific

literature, also to give a solution for the uncovered weaknesses from a cryptographical point of view.

The first section of the dissertation is an introduction. In the second one, those cryptographical foundations are formulated, that are essential for understanding the designed secure protocols. Cryptographical primitives, such as the encryption schemes, digital signatures, hash functions, and the timestamp are explained along with necessary preliminary mathematical knowledge, such as the finite fields, the random polynomials, and the Lagrange interpolation.

In the third section, two protocols are presented to solve the problem of authenticating the location information, calculated by a mobile device from the data transmitted by GPS satellites. The protocols developed by my coauthor and me are called „location-stamping". In addition to the description of the „location-stamping" methods, security analysis for the higher safety level protocol is given in Section 3. Furthermore, a US patent application of a portable electronic device is presented applying one of our „location-stamping" protocols. Additionally, in Section 3., the weaknesses of the GPS are analyzed extensively from the geographical and cryptographical points of view. Existing solutions for the well-known and recently detected security problems are also reviewed and presented here.

The fourth section deals with the security breaches of critical infrastructures. In this section, the detailed description of SCADA systems and industrial protocols are given, focusing on the Modbus protocols. Security problems and lack of security are analyzed by the "Attack Tree Method" to point out the weaknesses of the Modbus protocols. A secure Modbus RTU protocol is presented. The bibliography contains references in the order of appearance.

Summarizing the above, my dissertation aims to reveal and solve the cryptographic/structural shortcomings of two communication methods. Communication between the GPS satellites and receivers and communication between field devices (like sensors and actuators) in a SCADA system.

1 CRYPTOGRAPHICAL FOUNDATION

As it is mentioned in the introduction, this dissertation aims to reveal and give solutions for the cryptographic problems of two different types of communication methods, for the problem of GPS signal authentication and the problem of authentication of messages in a Modbus RTU flow. To understand the connection between the two parts of this dissertation, first, it is essential to know what is meant by **communication** in both cases. In the field of cryptography and signal processing the concept of communication defined by the Shannon model, which is called sometimes Shannon-Weaver model designed by Claude E. Shannon [3] and Warren Weaver [4]. The model can be seen in Figure 1. In a later section, the Shannon-Weaver communication model with extended information related to the communication in GPS and Modbus respectively is presented.

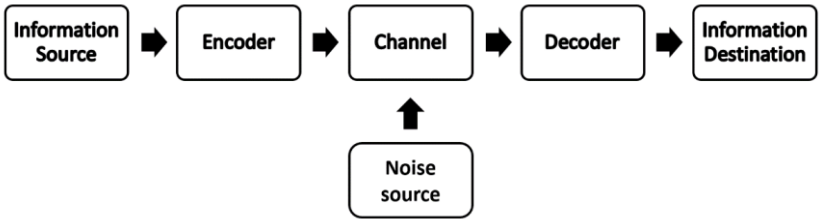

Figure 1. Shannon-Weaver model of communication

As Figure 1 shows, communication is always realized between an **information source** (sender) and an **information destination** (receiver). The information source is the part "which is producing a **message** or sequence of messages to be communicated to the receiving terminal" [3]. On the other side, the information destination is the part of the system which is intended to get the message sent by the information source.

The message can vary in type. For example, the time and destination information sent by a GPS satellite or a command sent in a Modbus based SCADA system, which is resulted in the change of some characteristic can be considered as messages. The information is transformed with the help of an **encoder** to a physical signal which can be carried on the communication channel, and back to its original form by a **decoder**. The encoder "is the party which operates on the message in some way to produce a signal suitable for transmission over the channel" [3]. The decoder "ordinarily performs the inverse operation of that done by the encoder, reconstructing the message from the signal" [3]. A **channel** is a physical medium, which can be the air, vacuum or any kind of metal together with a physical signal, for

example, a sound or an electromagnetic wave like radio, microwave, infrared, gamma transmitted by the medium.

The channel in the case of the GPS is the vacuum and air together with different radio waves or in the case of Modbus based SCADA systems the channel can be the air together with a radio wave especially along a metal wire namely wire waves.

In cryptography, **insecure** and **secure communication channel**s are distinguished. Unsecure communication channels are exposed to several different threats, so the communication and the transmitted messages can be damaged in many ways. If the channel is insecure, an attacker can read, modify or delete the messages according to the Dolev-Yao model [5], and the channel can be controlled by the attacker too. It means that many security properties of the communication can be violated. Reading the message harms the **confidentiality** property of the communication, modification of the messages harms **integrity** and **freshness** and controlled channel can harm both confidentiality and integrity furthermore it allows the attacker to impersonate the sender or the receiver in the communication, which harms the **authenticity** of the participants.

As a conclusion, it can be stated that insecure communication channels are lack of every property that is necessary to prevent the attacks against communication like eavesdropping, modification of the messages sent through the channel or impersonating the participants of the communication. To be able to provide reliable information exchange in any communication, a secure communication channel is necessary. Based on the related scientific literature [6], [7], [8], [9] a communication channel is considered secure if it provides the following properties between the sender (information source) and the receiver (information destination) during the transmission of the messages: confidentiality**,** cryptographic authenticity (from now on authenticity)**,** data integrity and freshness. Although such a channel will not guarantee, that the messages are ever received [10].

“Confidentiality and authenticity are independent but dual properties” of secure channels, as stated in [8]**.** Confidentiality can be defined in many ways; a few definitions are listed below. According to Mauer et al., confidentiality “intuitively, means that the encrypted message (the ciphertext) transmitted from the sender A to the receiver B does not leak information about the contents of the message (other than, for example, its length)” [11]. Another paper of Maurer et al. defines confidentiality as the following: “a channel provides confidentiality if its output is exclusively accessible to a specified receiver and this fact is known to the potential senders in the channel” [8]**.** Based on the book of Katz et al., confidentiality is “keeping information secret from all but those who are authorized to see it” [12].

Another definition by Blanchet says that “confidentiality, means that the adversary cannot obtain some information on data manipulated by the protocol” [13].

Like confidentiality, authenticity can be defined in many ways too, and a few definitions are mentioned below. Maurer et al. definition is the following “a channel provides authenticity if its input is exclusively accessible to a specified sender, and this fact is known to the receivers” [8]. Katz et al. say [12] that authenticity definition is the next: “authenticity requirements include knowledge or verifiability of the true identity of the party a key is shared or associated with.” Definition based on Blanchet’s paper is: “when B thinks he has run the protocol with A, he emits a special event end. When A thinks she runs the protocol with B, she emits another event begin. Authenticity is satisfied when B cannot emit his end event without A having emitted her begin event” [14]. Another definition by Blanchet says that “Authentication means that, if a participant A runs the protocol apparently with a participant B, then B runs the protocol apparently with A, and conversely. One often requires that A and B also share the same values of the parameters of the protocol” [14]. In this dissertation, the definition of Blanchet et al. [8] is used for both properties. It has to be remarked that authenticity in the above sense can be achieved by **message authentication** or **entity authentication**. “Message authentication simply authenticates one message; the process needs to be repeated for each new message. Entity authentication authenticates the participant of the communication (i.e. message sender and receiver) for the entire duration of a session.” [15]. Alternatively, “entity authentication mechanisms allow the verification, of an entity’s claimed identity, by another entity, and, the authenticity of the entity can be ascertained only for the instance of the authentication exchange” [16].

The next cryptographic characteristic that has to be considered is the data integrity, and while definitions of confidentiality and authentication can vary, data integrity is clearly defined in the related scientific literature. Data integrity is the property that provides proof that the message between the participants (message sender and message receiver) has not been modified throughout the communication. Alternatively, as Katz et al. give, data integrity is “ensuring information has not been altered by unauthorized or unknown means” [12]. As the last property the freshness is defined in the following way, “a given message is new, in the sense that it is not a replay of a message sent at a previous time” [17].

In cryptography when a communication channel has to be secured, the following methods are available: **symmetric** or **asymmetric** (or public) **key algorithms** and **cryptographical hash functions** (from now on hash functions). In this dissertation, only a brief explanation can be found about the cryptographic algorithms mentioned

above, for more detailed information see [12] and [17]. Figure 2. shows the provided cryptographic properties by the listed algorithms.

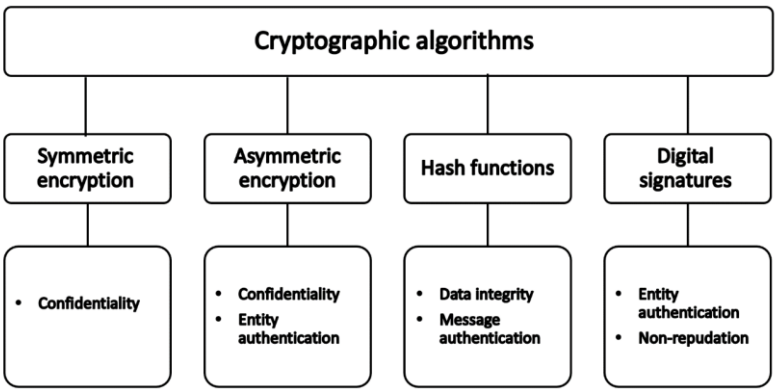

Figure 2. Provided cryptographic properties by cryptographic algorithms

To define the abovementioned cryptographic methods, it is necessary to introduce some of the basic concepts of cryptography. A **cryptosystem** is defined as a tuple of five sets, namely the **plaintext space**, the **ciphertext space**, the **keyspace**, the family of **encryption function**s and the family of **decryption function**s. The plaintext space consists of all the messages to encrypt in a language that is commonly understood, and the ciphertext is the set of all possible combinations of a given cipher alphabet. Keyspace contains keys to encrypt and decrypt messages. Encryption functions perform the process of encrypting the plaintext to the ciphertext to only authorized parties can access it. Similarly, decryption functions perform the process of taking the ciphertext and converting it back into plaintext. Both encryption and decryption functions require a key to operate correctly.

Before describing the existing cryptosystems, a few essential mathematical concepts and problems are defined and formulated here for better understanding. In cryptography for security considerations, many times huge numbers are used up to 4000 bits binary length. To be able to improve the speed of operations between these numbers, special sets are used, especially **finite fields**. Based on [12] the finite field "is a field F which contains a finite number of elements. The order of F is the number of elements in F. If F is a finite field, then F contains p^m elements for some prime p and integer $m \geq 1$. For every prime power order p^m, there is a unique finite field of the order p^m. This field is denoted by $GF(p^m)$, or F_{p^m} ." If $m = 1$ then it is called **finite prime field**, otherwise **extension field**. Elements of the finite prime field $GF(p)$ can be represented as integers of the set {0,1,…,p-1}, elements of extension fields can be represented as polynomials over this set. In a finite field, the elements can be added, subtracted, multiplied and divided each other with getting a result that exists in the finite field. A distinct element of a finite field is called the

generator element, which is exceptional in such a way, that all the elements of a given finite field can be calculated by raising the generator elements on different powers.

Elliptic curves are another particular set of points that can serve as a basis of hard mathematical problems. Elliptic curves are such algebraic curves, where the points on the curve satisfy the following equation : $y^2 = x^3 + ax + b$, where x, y, a, b are elements of a finite field. That points together with the infinite one form a finite Abelian group with the "chord and tangent" method. An elliptic curve can be defined over any finite field, although these curves have different characteristics.

Random polynomials are another important concept, that is used in cryptography often. Under random polynomial, those polynomials are meant that are having random coefficients from a uniformly distributed set of values. The random polynomials can be defined, for example in a finite prime field. That case is formulated such as $P(x) = \sum_{i=0}^{n} a_i x^i$, $a_i \in GF(p)$

As a basis of cryptosystems, hard computational problems are used, such as the **integer factorization** or the **discrete logarithm problem**. Integer factorization refers to the following. If a positive integer n is given, then the $p_i, i = 1 \dots k$ prime numbers need to be found if $n = p_1^{e_1} p_2^{e_2} \dots p_k^{e_k}$.

Discrete logarithm problem can be defined in finite fields or over elliptic curves. The problem is formulated in a finite prime field below. Given a prime p, a generator α of $GF(p)$ and β element of $GF(p)$ find the integer x, where $0 \leq x \leq p - 2$ and satisfies the equation $\alpha^x \equiv \beta \, mod \, p$.

After providing the definitions, it is possible to explain the cryptosystems used in this dissertation. Based on the number of the keys and the relation between the encryption and decryption keys, symmetric and asymmetric cryptosystems can be distinguished. In the case of symmetric cryptosystems, **symmetric key algorithm**s are used for the process of encryption and decryption, otherwise asymmetric key algorithms.

In symmetric-key cryptography, a **shared secret key** – encryption and decryption keys are the same – is used between the two parties of the communication, and with this common key, the messages can be encrypted and decrypted although the exchange of the shared secret key can be challenging, mostly because of the lack of a secure channel between the participants. Exchange of the common key on an insecure communication channel results in that the key will be revealed for any malicious third party. In addition, trust issues might arise between the participants of the communication. In the field of cryptography, it is called the **key distribution problem**. Thus, a secure key establishment method is essential. Under key

establishment based on [12] “a process or protocol whereby a shared secret becomes available to two or more parties, for subsequent cryptographic use” is meant. Key distribution problem can be solved by **key agreement protocol**s, **key transport protocols,** or by **secret sharing methods**, which all different approaches of key establishment. Key transport protocols are those, “where one party creates or otherwise obtains a secret value, and securely transfers it to the other(s)” [12]. While key agreements are those, in which two (or more) parties derive a shared secret as a function of information contributed by, or associated with, each of these, (ideally) such that no party can predetermine the resulting value” [12]. Secret sharing “involves a dealer who has a secret, a set of n parties, and a collection A of subsets of parties called the access structure. A secret-sharing scheme for A is a method by which the dealer distributes shares to the parties such that, any subset in A can reconstruct the secret from its shares, and any subset not in A cannot reveal any partial information on the secret” [18]. It can be stated, that without a trustworthy key establishment, the set-up phase of any symmetric cryptosystem is cumbersome. The most commonly used key establishment protocols are the **Station to station protocol** [18], the **Internet Key Exchange Protocol** family [19], the **MTI** family of protocols [20], the **MQV** protocol [21] or the **Shamir Secret Sharing** (SSS) [23] method and its alterations. The protocols are based on asymmetric key cryptography, explained later. SSS uses random polynomials over finite fields along with the Lagrange interpolation method to share and reconstruct keys between multiple parties.[1] The method is based on the fact that finding real roots of a high degree polynomial is hard if the degree is greater than 5. Lagrange interpolation[2] provides a way to reconstruct the secret. If the shared key cannot be revealed, then the confidentiality property of the secure channel is provided. However, these methods cannot assure the authenticity of the messages or the entities and the data integrity and freshness of the messages, because symmetric key algorithms are sensitive to **brute force**, **impersonation** and **man-in-the-middle attacks** (MITM) [22], [23]. So, additional technics like key agreement protocols have to be used because

[1] Participants: dealer and stakeholders.Goal is to divide a secret S into n shares $(s_0, s_1, \dots, s_{n-1})$, such that knowledge of t or more shares makes S easily computable, but knowledge of $t-1$ or less shares leaves S completely undetermined. Setup: the dealer chooses a large prime q, and selects a polynomial with the degree $t-1$ over $\mathbb{Z}_q^*$ such that $P(0) \equiv S\ (mod\ q)$. Then computes $s_i \equiv P(i)(mod\ q), i = 1, \dots, n$ and distributes s_i to the shareholders D_i , $i = 1, \dots, n$. Reconstruction phase: for any SH group of t shareholders compute the $P(0) \equiv \sum_{i \in SH} s_i L_i(0) mod\ q$, where L_i-s are nonsecret constants.

[2] Given a set of points $(x_i, y_i), where\ x_i \neq x_j\ i, j = 1 \dots t$, the Lagrange interpolation is defining a polynom, which roots are the given points. The polynom can be found in the form given below: $L(x) = \sum_{j=0}^{k} y_j l_j(x)$, where $l_j(x) = \prod_{\substack{0 \le m \le k \\ m \neq j}} \frac{x - x_m}{x_j - x_m}$, $j \in [0, k]$

of the beforementioned manner to reach the goal, namely, to provide a secure communication channel.

Symmetric key algorithms can be classified into **block** and **stream cipher**s based on the amount of data encrypted at a particular time. Block cipher encrypts a predefined block size while stream ciphers one bit or byte of the plaintext at one step. In the case of stream ciphers, the used keystream has to meet strict requirements, to ensure a proper security level. Depending on the underlying block coding mode, the block ciphers are better in fixing noise or attacker generated modifications during decryption. Block ciphers are more popular and widespread.

The most commonly used symmetric key algorithms are the **Triple Data Encryption Standard** (**3DES**) [24], the **Advanced Encryption Standard** (**AES**) [25], the **Blowfish** [26] the **Twofish** [27] and the **Rivest cipher** family (RC). Members of the RC family are stream ciphers, the 3DES, AES, Blowfish, and Twofish ciphers are block ciphers. 3DES, Blowfish, and Twofish algorithms are **Feistel ciphers** (FC), and AES is a **Substitution-Permutation network** (SPN). Both FC and SPN uses more rounds to permute the input and invert the bits of the input, but that input is divided into N pieces in the case of the Feistel ciphers – $N = 8$ in 3DES, $N = 4$ in Blowfish and Twofish –, while the SPN operates on the whole input in every round. In Figure 3., the general structure of symmetric key algorithms is displayed.

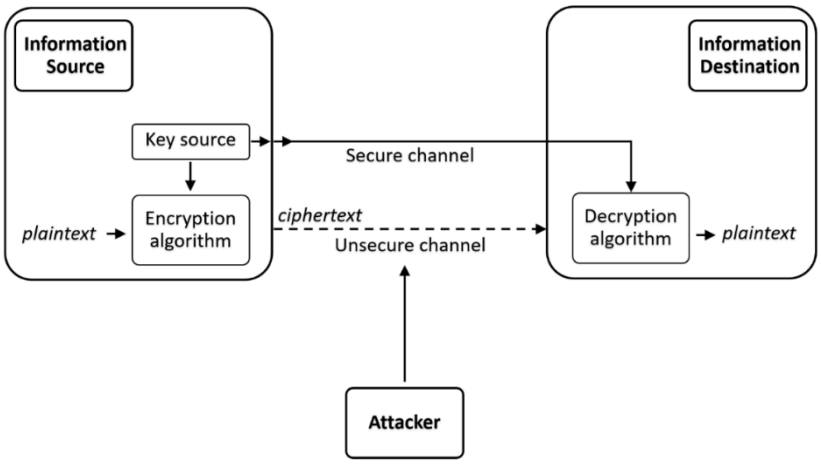

Figure 3. The general structure of symmetric key algorithms

Until the use of symmetric key algorithms mostly limited to data encryption, the asymmetric key algorithms used for more purposes, like data encryption, key-pair generation, digital signature or key exchange, in the case of **asymmetric key algorithm**s, a pair of a **secret/private** and a **public key** is used to ensure the confidentiality and entity authentication during the communication, while these

algorithms do not ensure data integrity and freshness. Private and public keys are different but mathematically connected. Only the private key must be kept secret, and there is no need for key exchange via a secure channel. Asymmetric cryptography is based on hard mathematical problems, like integer factorization along with modular exponentiation or discrete logarithm problem over different algebraic structures or elliptic curves.

The most commonly used asymmetric algorithms for key exchange, as it is mentioned before, the **Station to station protocol** [18], the **Internet Key Exchange Protocol** family [19], the **MTI** family of protocols [20] or the **MQV** protocol [21]. Station to station and Internet Key Exchange protocols are based on the hardness of integer factorization, MTI, and MQV protocols security is related to the problem of discrete logarithm.

The most often used asymmetric algorithms if the purpose is encryption are the following: **RSA** [28], **Rabin** [29] , **ElGamal** [30] and **ECC** (Elliptic Curve Cryptography) [31] [32]. The RSA algorithm is one of the first and still in use an asymmetric key algorithm that is ever designed, it is based on the hardness of integer factorization formulated earlier. RSA is traditionally used in TLS (Transport Layer Security) handshakes, or it is part of the long been popular PGP (Pretty Good Privacy) end to end email encryption program. Rabin encryption is the least popular, but still used encryption system of the above, its security is based on the difficulty of integer factorization also. The security of the ElGamal encryption scheme is related to the intractability of the discrete logarithm problem over a finite prime field explained previously. ElGamal encryption is modified by Koblitz [31] to use elliptic curves, and it is called **Menezes-Vanstone Elliptic Curve Cryptosystem** or **Elliptic Curve Elgamal** [33]. The ECC security relies entirely on the hardness of the discrete logarithm problem over elliptic curves. Elliptic curve cryptography is widely applied, for example, in the internal governmental communication of the USA, in the Tor browser or the technology of bitcoin. The leading role in the field of asymmetric cryptography is slowly taken over by elliptic curve cryptography. The general structure of asymmetric data encryption algorithms is displayed in Figure 4.

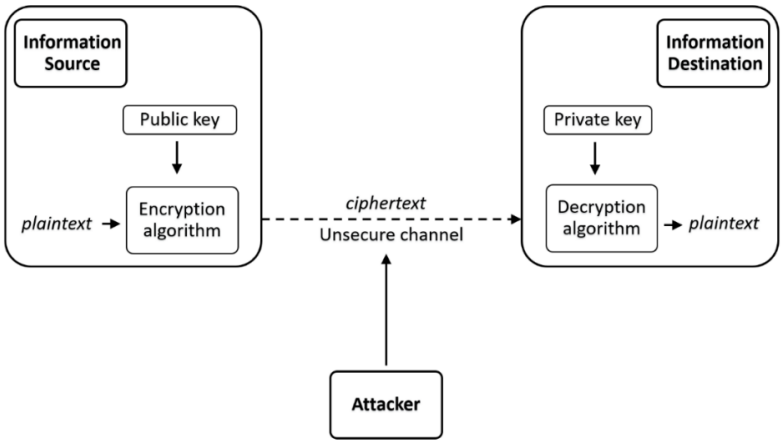

Figure 4. The general structure of asymmetric key algorithms

As it is mentioned above, asymmetric key cryptography is used to provide **digital signature**. In the case of the digital signature, the sender has to be authenticated in the communication. The sender using its private key and signs the message, on the other side, the receiver using the public key paired to the private key of the to be authenticated sender verifies the signature. Digital signature schemes are sensitive for MITM attacks and not able to provide freshness and confidentiality during the communication, but these algorithms ensure entity authentication and non-repudiation. The most commonly applied digital signatures are the following **RSA** (**Rivest-Shamir-Adleman**) [28], **Rabin** [29], **ElGamal** [30] and **Elliptic Curve Digital Signature Algorithm** (**ECDSA**) [34]. Although asymmetric key cryptography requires less effort when the secrecy of the keys is under observation, but the underlying mathematical methods and its implementation take much more time. The general structure of the digital signature can be seen in Figure 5.

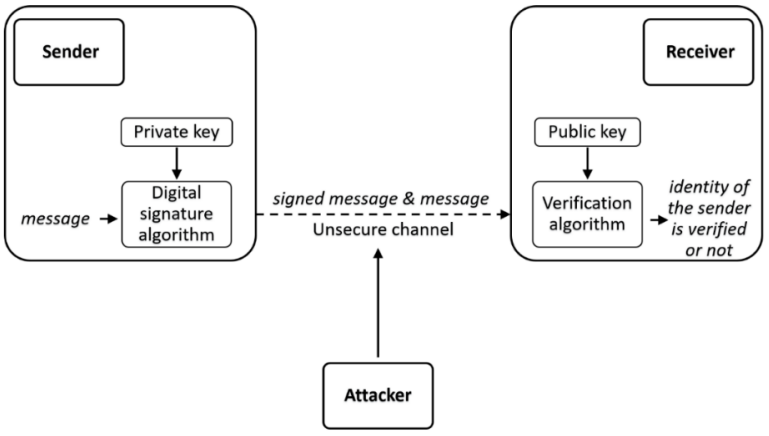

Figure 5. The general structure of digital signature algorithms

In the case of asymmetric key algorithms, there is no need to exchange keys because anybody can know the public key. Thus a way has to be provided to discover public keys and ensure the falsification and abuse of them. Infrastructure has to be set up, which is responsible for key distribution and key management; it is called **Public Key Infrastructure** (PKI) [35]. Trusted third parties are involved in the communication by the PKI, and certificates are provided by the trusted third parties to verify the connection between the public keys and the claimed owner entities. PKI is formed by several components, such as the **Certification Authority** (CA), the **Registration Authority** (RA), the **Certificate repository** (CR) and the **Key Archival Server** (KAS). Every user is assigned to a CA through a registration process maintained by the RA. Verifiable relevant information about the user is provided by the user during the registration, what then verified and associated with a unique username by the RA. Public-private key pair is generated after registration in the PKI. The public key is then available to anybody, and the CA provides a proof – a certification – of the association between the identity of the user and the public key. In the PKI then the CA "acts as the root of trust in public key infrastructure and provides services that authenticate the identity of … entities in a network" [36]. Valid certifications are stored in the CR, after expiration time certifications are relocated to the KAS.

Hash functions as the third option, are one-way functions with an arbitrary length input and a fixed-length output. Hash functions are in possession of the properties such, the output is easy to compute for any given input, it gives the same output for one particular input (deterministic), the output is not revealing any information about the input (preimage resistance), and it is practically impossible to find two different inputs to get the same output (collision resistance). There exist two main types of hash functions, the **unkeyed**, which is used for providing data integrity; it is called **Message Integrity Codes** (**MIC**). Moreover, the **keyed**, which is able to provide not just data integrity but also message authentication, it is called **Message Authentication Codes** (**MAC**). Unkeyed hash functions have the advantages that no key is necessary, but only message integrity is provided, which is just one of the many properties that are necessary to ensure a secure channel. Keyed hash functions provide message authentication beside message integrity; that is the reason why these algorithms serve as a basis of several solutions. The most commonly used MIC are the members of the **Secure Hash Algorithm family** (**SHA**) [37], the **SHA-256** and **SHA-512** and the **Paralellizable MAC** (**PMAC**) [38]. The general structure of hash function can be seen in Figure 6.

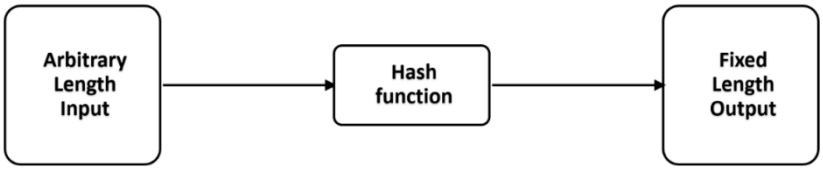

Figure 6. The general structure of hash functions

All the algorithms mentioned above referred to as **cryptographic primitives**, that are well established, comprehensively verified building blocks of cryptographic communication protocols.

Although the following – lastly mentioned – cryptographic primitive is not able to encrypt the plaintext, decrypt the ciphertext, sign or verify documents. In the communication process to check the freshness of a message, it is necessary to know the time of the sending and the receiving. To provide time information, the **timestamp** method is used in cryptography. It can be used with the time information of the participants, but the time provided by the parties cannot be assured as reliable information. To get reliable and useful time information, the clocks on each side of the communication have to be synchronized and secured. The timestamp guarantees reliable time information, timeliness, and uniqueness if the secure and synchronized clocks are provided, or a trusted third party generates the timestamp. Timestamps' function is to detect replay of messages. The exact definition is the following “timestamping is recording the time of creation or existence of information” based on Katz et al. [12] Timestamps are based on the idea, that if a message has arrived within a predefined time interval, then the message is fresh enough.

In this dissertation, a new type of cryptographic primitive a so-called **„location-stamp”** is introduced. Reliable time information is provided by the timestamp with „location-stamp” reliable location information – based on the GPS – is desired to provide. Referring back to the definition of the timestamp the „location-stamp” can be defined the following way: „location-stamping” is recording the location of and information at a given time.

2 GLOBAL NAVIGATION SATELLITE SYSTEM

3.1 GENERAL DESCRIPTION AND APPLICATIONS

3.1.1 GENERAL DESCRIPTION

"**GNSS** (**G**lobal **N**avigation **S**atellite **S**ystem) is a generic term denoting a satellite navigation system. A GNSS involves a constellation of satellites orbiting Earth, continuously transmitting signals that enable users to determine their three-dimensional position with global coverage" [39].

The story of GNSS – displayed in Figure 7. – started on the 4th of October in 1957, when the USSR launched the first satellite named Sputnik 1.

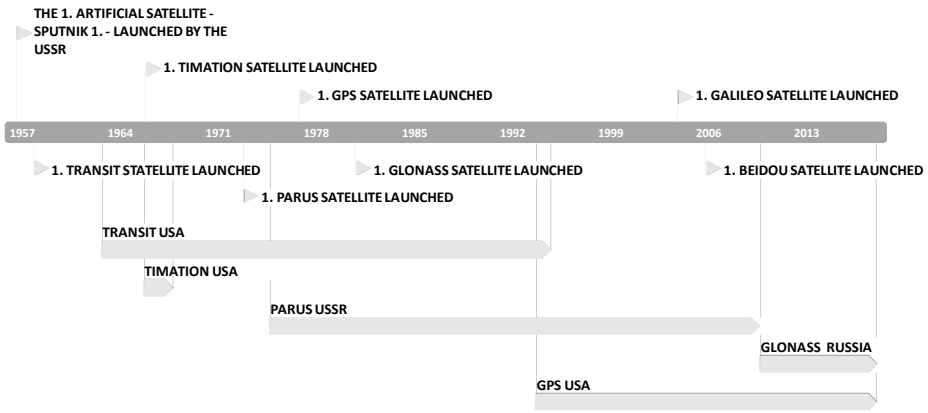

Figure 7. History of GPS and GLONASS until 2013

American researchers from the John Hopkins University started to investigate the radio signals emitted by the above satellite and found out that using the Doppler effect, they can determine the orbit of the satellite. Later, they figured out that their former method can be used to detect the location of any object on the surface of Earth. Soon the TRANSIT (Navigation Satellite Program) project in the USA and the PARUS project in the Soviet Union started, which aim was to launch satellites into space in the right constellation to be able to detect the precise global position of objects. In the 1970s during the so-called Space War, the **GPS** (**G**lobal **P**ositioning **S**ystem) and the **GLONASS** (**GLO**balnaya **NA**vigatsionaya **S**putnikova **S**istema) systems design process were started based on the experiences of TRANSIT and PARUS projects, improving the precision of the earlier ones by installing atomic clocks on the satellites. With the help of these exact onboard clocks, not only the accurate location but the precise time information can be provided as well. During the next two decades, the GPS improved a lot, and in 1995 it reached Full Operational

Capability (FOC). “FOC status means that the system meets all the requirements specified in a variety of formal performance and requirements documents” [40]. The GLONASS reached its FOC status much more later in 2011. Other nations installed two other GNSS in the last decade, the **BeiDou** Navigation Satellite System designed by China, and the **Galileo** (European Global Satellite Navigation System) by the European Union, both will reach the FOC state at 2020. It can be stated that today the positioning, navigation and timing services of the GNSS mentioned above together cover the whole globe. In the following, the technical description of the American GNSS, i.e. the GPS will be only given since at the time of our research it was the most popular, commonly used and stable system.

GNSS, in general consists of three main parts, the **Space**, the **Operational Control** and the **User Segment** as it is shown in Figure 8.

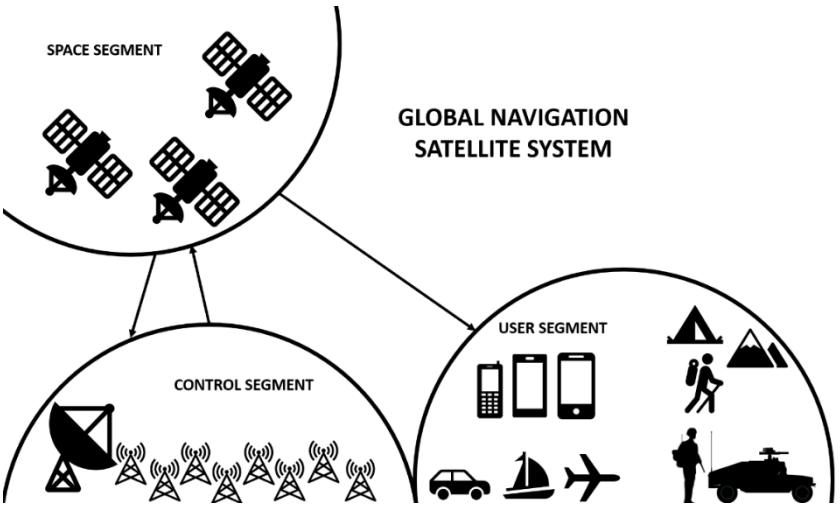

Figure 8. Structure of GNSS

In general, the Space Segment contains a set of satellites orbiting Earth in a **Medium Earth Orbit** (**MEO**). MEO is the area in space around Earth, which altitude from the sea level is higher than 2000 km but less than 35786 km. The orbital altitude of GLONASS is 19100 km, that of BeiDou is 21150 km, and that of GALILEO is 23222 km. Space Segment is the part of the GNSS, which is responsible for the generation, modulation, and transmission of code and phase signals. Tasks are controlled by 1 to 4 highly stable atomic clocks installed on each satellite. Furthermore, satellites of the Space Segment must store, periodically refresh and broadcast the **navigation message** (**NAV**) which is provided by the Operational Control Segment. “The NAV provides all the necessary information to allow the user to perform the positioning service” [41].

The Space Segment of GPS consists of 31 satellites (April 2019) [42], 24 satellites of them form the core constellation, the additional seven satellites can increase the

performance, reliability, and precision of the GPS. All the satellites are orbiting Earth in a MEO, about 20200 km high above the sea level, and with an orbital radius 26600 km. The 24 satellites are divided into six groups, and each group is located on different orbital planes. Orbital planes are inclined 55° to the equatorial plane, and the orbital planes are equally spaced about the equator with a 60° difference. Such a constellation can provide that any time anywhere on the surface of Earth at least four satellites is “visible” for a user, which is indispensable to get precise location information. Figures 9. and 10. are for a better understanding of the structure of the system. Table 1. shows the basic parameters of the GPS constellation. Data for this section are provided by [43].

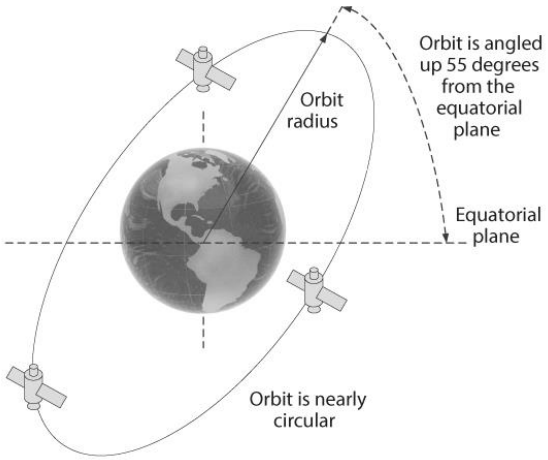

Figure 9. A GPS satellite orbit [44]

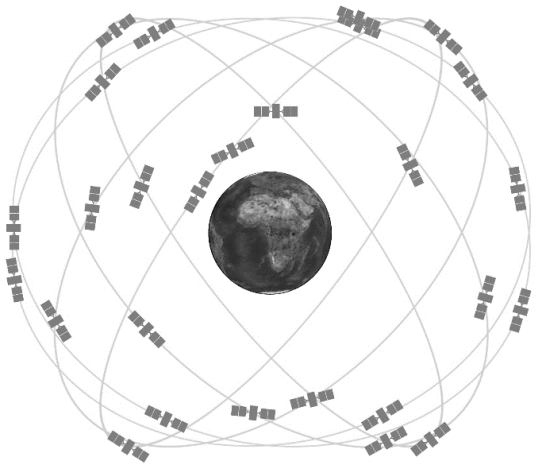

Figure 10. Expandable 24-Slot satellite constellation of GPS [45]

Table. 1. Parameters of GPS constellation

Parameter	Value
Total number of launched satellites	67
Number of active satellites	31
Number of inactive satellites	36
Number of satellites in the core constellation	24
Number of satellites in the expandable 24-slot constellation	27
Number of orbital planes of the constellation	6
Number of satellites on an orbital plane	4
The inclination of the orbital planes	55°
Orbital radius	26 600 km
Orbital altitude	20 200 km

Generally, the Operational Control Segment is the part that is responsible for the flawless operation of the constellation of the satellites and the generation and refreshment of the navigational message (NAV) transmitted by the satellites. For the purposes described above, it tracks the orbital configuration, checks the condition and the clock information, if it is necessary gives corrections for all these critical properties, and periodically resends the navigational message to all orbiting satellite.

In the case of the GPS, the Operational Control Segment contains one **Master Control Station (MCS)**, an **Alternate Master Control Station (AMCS)**, **Monitoring Station**s all around the world near to the equator and **Ground Antenna**s as it is shown in Figure 11.

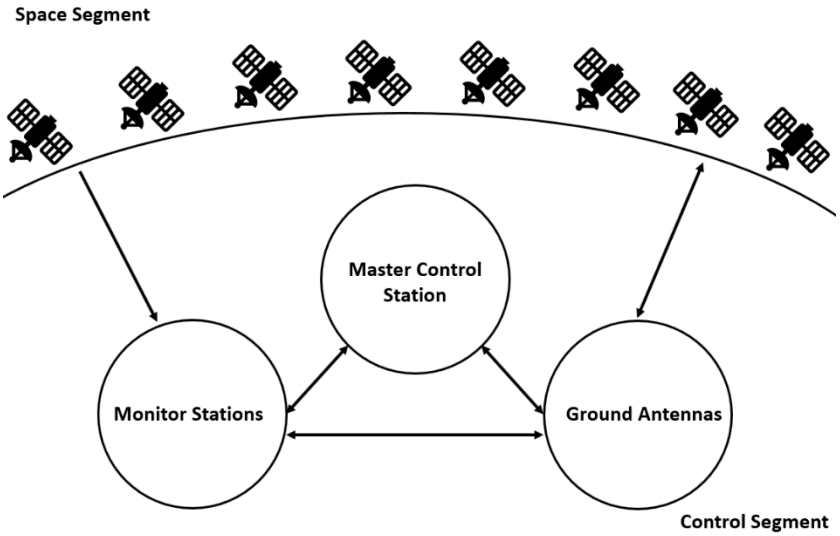

Figure 11. Control Segment

The MCS and AMCS control, maintain and operate the Space Segment, so generate the NAV, check the health of the satellites, synchronize the constellation and monitor the system performance. The Monitoring Stations have all the necessary devices to be able to collect and process satellite, atmospheric, and meteorological data. The Ground Antennas are sending the data – collected and processed by the other two parts of this segment – to the Space Segment to keep the system moving and to improve the effectiveness of it, using radio waves. Between the Space and the Ground Antennas the communication is two-way, but between the Monitoring Stations and the Space Segment communication is one way, since only the ground antennas can send and receive messages from the Space Segment. For a more detailed description of the system, see [43].

The third central part of the system is the User Segment which is composed of uncountable GNSS/GPS receivers. The main task of these devices is to receive the signals broadcasted by the satellites possessing the relevant information to solve the navigation equation in purpose to get the precise location information, i.e. to calculate the GNSS/GPS coordinates. As it is mentioned before the primary aim of the GNSS is to provide location information globally, but it can be used to many other purposes depending on the user's needs. The three most common applications are positioning, navigation and time determination (PNT). It is essential to mention that the communication between the User Segment and the Space Segment is one-way, the satellites broadcast information and the receivers are only able to receive it.

As it is stated earlier, the GNSS satellites are continuously broadcasting information. The type of carrier signals, the used radio frequencies and the content of the message sent through the air are depending on the specification of the different GNSS systems. In the case of GPS, two kinds of signals can be distinguished, the legacy carrier (before 2005) and the modern signal (after 2005). The legacy signals are transmitted via two radio frequencies named Link1 (L1) and Link2 (L2). Both frequencies are derived from a base frequency which is generated by an onboard frequency synthesizer device on each satellite, exact frequency of L1 is 1575.42 MHz, and L2 is 1227.60 MHz The modern signals are using the L1 and L2 carrier frequencies of the legacy signals, and a new carrier frequency the L5, which exact value is 1176.45 MHz. Two services are provided by the GPS, namely the **Standard Positioning System** (**SPS**) and the **Precise Positioning System** (**PPS**). SPS is for public use, it is open and free of charge, in contrast to PPS, which is an encrypted service available only for military and authorized civil users. In the case of the legacy signals, the L1 carrier is used for transmitting the SPS and PPS, but L2 is only reserved for the PPS. Because the PPS is capable of dual-frequency mode, the accuracy of this service is higher than the accuracy of the SPS, namely the overall global average accuracy of SPS is 7.8 m while this value is 2.6 m in the case of PPS. L1 and L2 are modulated with

different pseudo-random-noise (PRN) codes to be able to carry the **course/acquisition code** (**C/A**) for the civil, and the **precision code** (**P(Y)**) for the military users respectively. In the case of modern signals, the L1, L2 and L5 carrier frequencies are used for SPS and L1, and L2 are used for PPS. L1 is modulated with different PRN codes to carry C/A, L1C, P(Y) and M-codes, after modulation L2 can carry the L2C, P(Y) and M-code, and L5 with the help of it.

Accuracy of the recently introduced L1C, L2C, and M-code matches or slightly outperforms the performance of the legacy signals. Earlier because of safety reasons the so-called **Selective Availability** was applied on both services, i.e. both codes were encrypted with a time-varying frequency offset, and the ephemeris and almanac information were encoded too. Selective Availability was terminated in 2000. Legacy GPS signals are carrying the **GPS Navigation Message** (**LNAV**), LNAV messages contain information about the health of the satellite, ephemeris and almanac parameters, ionospheric parameters, clock correction terms and satellite configuration parameters. Ephemeris and almanac describe the orbital parameters and clock information of the actual satellite, the almanac is less accurate than ephemeris, and GPS coordinates can be calculated without the almanac, but ephemeris is essential for the calculation. Modern GPS signals are containing the CNAV or the CNAV2 message, which in content only slightly differs from the legacy NAV, but the structure is different in many ways. The following table shows the characteristics of the SPS and PPS services.

Table. 2. GPS signals

	Standard Positioning Service	**Precise Positioning Service**
Application	Civil	Military
Legacy		
Frequency	L1	L1, L2
Signals	Coarse/Acquisition (C/A)	Precision code (P(Y))
Modern		
Frequency	L1, L2, L5	L1, L2
Signals	L1C, L2C, L5	M-code

The most often used service of GPS is positioning, which is for determining the position of a given object on the surface of the Earth. If the reference system is known, then the position can be given by three coordinates, which are the latitude, the longitude and the altitude coordinates of the object. Based on the related literature usually, two base reference system is used in the case of GPS, the **Conventional Celestial Reference System** (**CRS**) and the **Conventional Terrestrial Reference System** (**TRS**). As the following table shows the CRS is an inertial reference system, which origin is located in the Earth's center of the mass, the x-axis of it is

heading to the vernal equinox, and z-axis of it is orthogonal to the equatorial. TRS system origin is the Earth's center of the mass, the x-axis in this particular system is heading to the intersection of the equatorial and mean Greenwich meridian, while the z-axis is equal to the axis of the Earth's rotation and it is heading to the Conventional Terrestrial Pole. Y-axis in both coordination systems can be found like complementing the right-handed system with the corresponding x and z axes. The differences between the two systems include not only the technical parameters but the purpose of the systems as well. CRS is used to define the location of the satellites in the system precisely. TRS is for defining the location of a receiver.

Table. 3. Reference frames

Coordinate system	**Type**	**Origin**	**X-axis**	**Z-axis**	**Y-axis**
CRS	Inertial	Earth's center of the mass	Heading to the vernal equinox (first point of Aries)	Orthogonal to the equatorial	Complementing the right-handed system together
TRS	Earth-Centered Earth Fixed	Earth's center of the mass	Heading to the intersection of the equatorial and mean Greenwich meridian	Same as the axis of Earth's rotation and it is heading to Conventional Terrestrial Pole	Complementing the right-handed system together

So, if the GNSS/GPS receiver is available, and the satellites are broadcasting the proper signals included the ephemeris of the system, then the positioning can be done. The receiver at first starts to look for visible satellites in an area of the sky 15° above the horizon. Based on the actual time of the receiver, the almanac of the system – which, as it was mentioned before contains information about the orbit of the satellites – the receiver can easily find the three or four satellites which are at least necessary to the positioning method. Ephemeris is containing information – like clock correction values, more precise orbital data – which helps to improve the accuracy of the positioning. The positioning method is based on a well-known mathematical principle, namely on the trilateration. Applying trilateration, the position of an object near the surface of Earth can be determined as follows:

1. The receiver gains signal from three satellites.

2. Calculates the distances – not the accurate geometric distances – between itself and the satellites. The calculation is based on the travel time of the signal from the satellite to the receiver. The travel time (ΔT) is the difference between the emission $(t^{sat}(T))$ and reception time $(t_{rcv}(T))$.

$$\Delta T = t_{rcv}(T) - t^{sat}(T)$$

   The travel time is then multiplied by the speed of light to get the pseudorange (R) between the satellite and the receiver. The gained distance is called pseudorange because although the emission time is measured by a high precision atomic clock installed on the satellite, the inaccurate clock of the receiver provides reception time; thus the pseudorange contains many errors.

$$R = c\Delta T$$

3. The three pseudoranges then determine three spheres on which the receiver can be located. The intersection of these three spheres determines two points, but only one of them is located somewhere near to the surface of Earth, and this is the one considered as the position of the receiver. Trilateration presented in Figure 12.

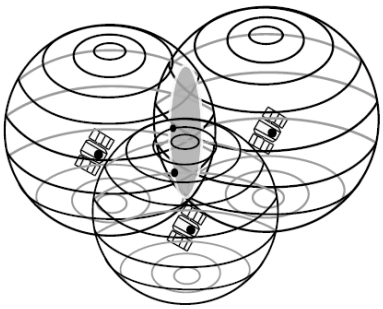

Figure 12. Trilateration [43]

The position determined the way as mentioned above, with the help of three satellites, is accurate only if the clocks of the satellites and the receiver are precisely synchronized, which is usually not true. To eliminate the error caused by clock problems, a fourth satellite is necessary. Then the method is the following:

1. The receiver gains signal from four satellites.

2. Calculates the four error-laden pseudoranges (R_i) between itself and the satellites. Since the same receiver is used to calculate the pseudoranges, the

error in the reception time (t_{cor}) is equal for all observations. The clock error then can be calculated, and the measurement can be corrected.

$$\left.\begin{aligned} R_1 &= c(t_{rcv}(T_1) - t^{sat}(T_1) + t_{cor}) \\ R_2 &= c(t_{rcv}(T_2) - t^{sat}(T_2) + t_{cor}) \\ R_3 &= c(t_{rcv}(T_3) - t^{sat}(T_3) + t_{cor}) \\ R_4 &= c(t_{rcv}(T_4) - t^{sat}(T_4) + t_{cor}) \end{aligned}\right\}$$

3. The intersection of the four spheres determined by the corrected pseudoranges defines a single point, which is the desired position.

3.1.2 APPLICATIONS RELATED TO MY RESEARCH

While the years have passed since the day, when Bill Clinton president of the USA has ordered to stop disarrange the satellite signals, the number of services based on the GNSS/GPS has increased in an unexpectedly considerable amount. Only creativity and imagination can give a limit to the different types of applications. The original purpose of the GPS was clearly military, but after the **Selective Availability** was terminated such an opportunity gained ground which caused significant changes in many fields of the civil, scientific, industrial and governmental segment as well. In the following, a non-completed list of the available applications is given (see Table 4). Such an application became a vital part of our everyday life. Based on a survey, which was taken in 2015 by the Geo awesomeness [46], 90 percent of people use applications based on GPS on their mobile phones in the USA. So, it seems obvious researching the authentication issues of this field.

Providing cryptographic or structural protection of any data – like any other safety solution – costs time, storage place and money, so only such data should be protected, which are valuable enough. The question arises: are these GPS coordinates so important? The answer depends on the application. For example, in the case of navigation, the GPS receiver calculates GPS coordinates so often, and the calculation of the coordinates takes a little time only, and if it is a public application, protecting the data cryptographically is not very important, on the other hand, the structural safety of the service is necessary. However, for example, if proof of the location of your neighbour's fence is needed, in order to prove that it is falsely placed on your property, then it is quite sure that the official establishments are expecting more reliable proof than the GPS coordinates of the fence calculated by a simple GPS receiver. To get satisfying proof for the above problem, exact measurement is necessary proceeded by the Land Register. Processing of such a complaint takes much time, but if reliable – approved by the authorities – GPS coordinates would be available, time and money could be saved. Considering the following situation reliable GPS coordinates provide a suitable solution too. If a car is parked on the border of a parking zone, where parking is free of charge, and a mindless parking

attendant imposes a fine still even though there is no misdemeanour happened, a reliable proof is needed to protect ourselves. Mobile phones are always available and can calculate GPS coordinates, i.e. if an application would exist which provides reliable GPS coordinates. The only thing which should be done is installing the application on the mobile phone and calculating the coordinates of the car then file the information to the authorities. Another possible situation, where reliable GPS coordinate can be helpful too, is the following. A supervisor of the Land Registry founds a field with a serious proliferation by ragweed, and then the supervisor fines the owner of the property. To create a proof, an official note is made in such steps:

1. A mobile device calculates GPS coordinates of the area.
2. Puts the GPS coordinates into the note.
3. The officer signs the note digitally.
4. Furthermore, asks for an authentic time stamp.

With the steps mentioned above, the identity of the supervisor, and the maker of the note is proven.

If a legal procedure is necessary in any of the above cases, then neither the penalized driver nor the supervisor can prove that their GPS coordinates match the place where they made it, despite the fact, that the supervisor certified the location information by his digital signature. They cannot prove that the location information is correct because not a single person or device is fully trusted too. Electronically saved or transmitted data can be changed easily, thus without any kind of provable signature, the reliability of data is always questionable.

In this dissertation, the focus is on those types of applications where the received GPS coordinates have probative value in a legal procedure or dispute. The Police Service makes such an application of Northern Ireland, called the **Noise App** [47], which, as the developers state, can be used to record nuisance at a precise moment it is occurring. It promises to provide evidence in cases when a legal procedure is needed to resolve the situation. Typically, to create a record about such an event, a policeman's contribution is necessary. Nevertheless, if the Noise App is used, then it allows you to file the problems without any help of an investigating officer. The application notes not only the recordings you have made but the exact time and location information provided by the GNSS.

Another example can be taken from the field of logistics. For the shipping companies, it is crucial to be able to prove that the delivery has been completed in the right condition, at the right time, to the right location. Usually, shipping companies use some tracking applications based on the GNSS to provide proof of delivery, but when they try to prove that the packages are intact and unharmed in the time of shipment

some additional information is necessary, like a photo, which is one of the many available features that **myGeoTracking** [48] provides. With the help of myGeoTracking, a GPS tagged photo is captured for the proof of delivery. The GPS tagged photo holds the time and the location information of the delivery and the condition of the package at that exact time and location. Later, as the developers promise, these photos can be used as evidence in the case of customer complaints.

The following compelling example is connected to the field of law enforcement. There exists an application named **Alibi** [49], which can confirm your location with the help of some biometric data and GNSS. Alibi, as the developers state it, is capable of saving and authenticating the exact location and time information of the user arriving at a specified destination at a specific time, based on a biometric data, especially the fingerprint. The fingerprint is added with the help of Apple's TouchID tool of the device. Although, it seems a little bit odd to think ahead so far, but if you need an alibi, in the case of any accusation – which can be made by for example your boss, your spouse or in the worst scenario by the police – it can be accommodating to prove your innocence.

The example below is presenting the situation which tries to be solved by the protocols introduced in this dissertation. Imagine that you are driving home after an exhausting workday, and you are not noticing the traffic sign, which shows that the speed limit has changed to 70 km/hour. You are driving through without slowing down, but unfortunately after the next turn of the road, from the cover of a bush, a policeman appears with a radar. Furthermore, a few days later you get a speed ticket – usually a photo as you are exceeding the speed limit – with exact time and location information about the event, and then you have to pay the penalty, without any excuse. Using an authenticated device, with a built-in GPS receiver, the police officer has the authority to penalize you since it proves that you were at that certain location at that certain time.

All the applications mentioned above provide data to prove something undeniable to another party, like nuisance recordings provided in the case of Noise App or speeding with the help of speed radars, or completed shipping regarding myGeoTracking application, or alibi applying the application Alibi. As it can be seen the common part in all examples presented above, is that a GNSS provides the authenticity of the time and location information. GNSS data are considered reliable and valid by all the applications. However, it has to be emphasized that none of the GNSS ever promises that all the receivers will work fine, without any mistake or miscalculation, or malicious intervention by an attacker. The NAV is the only information, which can be guaranteed to be accurate. In the next section, all the related weaknesses of the GNSS are discussed.

3.2 OWN RESEARCH

3.2.1 REVEALED WEAKNESSES

Even though the number of GNSS based services, especially the GPS based ones has been increased remarkably in the last decades, only a small part of the system's security issues got enough attention in the related scientific literature. Through these years the GNSS technology has gained a considerable portion in the civil, scientific, industrial, governmental or military sectors, and the list is not complete at all. As can be seen in Table 4. the applications are providing solutions for many types of problems, but if the consequences of the lack of cryptographic and physical safety are not considered sufficiently, it could lead to severe problems. The occurring security issues or vulnerabilities are mostly related to the stable operation of the GNSS itself, such as guaranteeing uninterrupted service or providing more authentic geographic data, and obviously, the malicious attackers should not be forgotten as well.

Table. 4. Applications of GNSS/GPS

Type of application	Field of application	Examples
Civil	Navigation	guided tours, geocaching
	Location-based applications	exercising, dating, gaming, marketin, etc.
Scientific	Surveying	mapping seafloors determining land boundaries
	Wildlife tracking	tracking the movement of wild animals
	Seismic research	measuring the motion of plates related to each other
Industrial	Agriculture	precise farming, auto-driving tractors
	Power grid	phase synchronization
	Transportation and navigation	car, railway, marine, aviation
Governmental	Emergency services	enhanced 911
	Legal and Law Enforcement	geo-fencing, fishing zone, taxation, penalty
	Timestamp	financial, clerical, legal
Military	Defence	munition guidance
	Search and rescue	
	Unmanned vehicle	

The original aim of the GNSS was clearly military, and in the time when the first GNSS, the GPS was launched the only security issue that was attractive to the developers was to protect the geodetically authenticate position data from the unauthorized users and keep the service stable and reliable to provide position, navigation and time information that can be used safely for military operations. The problem of

unauthorized access has been solved, with the help of the Precise Positioning Service, which used an encryption method on the broadcasted signal to prevent the unbidden access. Precise Positioning Service was transmitted the P(Y) code on both the L1 and L2 carrier signals of the GPS. The so-called Selective Availability which hides the precise position data from non-military users was discontinued in 2000, and when the termination of it came into effect, the only cryptographical security was disappeared from the SPS. Based on the technical literature available [39], [41], [42], [44], [43], the communication between the User and Space Segment has no other cryptographically secured part. The vulnerability of the GNSS was examined continuously through the last twenty years, by several research groups and official departments [50], [51], [52], [53], [54], [55], [56], [57]. The reliability of the system is based on two factors; the underlying technology, and the communication methods, so the obvious thing is to supervise these two factors. The technology underlying the GNSS/GPS has developed a lot through the last half a century. A significant change can be realized at the satellites if the technical parameters are compared. There are differences in the type of the used batteries, onboard clocks, electric power of the solar panels, weight, length of the predicted and actual lifetime. However, not only the design of the satellites and the inbuilt accessories has changed, but the used radio frequency, the broadcasted signals and the way of communication have undergone fundamental changes too. The improvement in the degree of geodesic authenticity was an important factor as well. In Table 5, the evolution of the used satellites can be seen in the case of GPS and Galileo. The development aimed to improve the lifetime of the satellites, create such devices which maintenance is more comfortable, cheaper and form a solid base for the positioning service, and finally make the service more resistant to the known weaknesses of the system.

Table. 5. Satellite specifications

	Launch date	**Solar panel**	**Onboard clock**	**Design lifespan**	**Weight**
GPS					
Block I	1978-1985	800 W	1Cs 3Rb	4.5 years	759 kg
Block II	1989-1990	800 W	2Cs 2Rb	7.5 years	840 kg
Block IIA	1990-1997	800 W	2Cs 2Rb	7.5 years	1500 kg
Block IIR	1997-2009	800 W	3Rb	7.5 years	2000 kg
Block IIR-M	2005-2009	800 W	3Rb	10 years	2032 kg
Block IIF	2010-2016	1952 W	1Cs 2Rb	12 years	1633 kg

	Launch date	Solar panel	Onboard clock	Design lifespan	Weight
Block III	2018-	4480 W	3Rb 1H	15 years	3880 kg
Galileo					
	2011	1.9 KW	2H 2Rb	12 years	700 kg

3.2.1.1 WEAKNESSES OF GPS RELATED TO GEODESIC PROPERTIES

To be able to define the concept of geodetic authenticity, the concept of accuracy and precision have to be defined first. "The **accuracy** of a measurement is how close a result comes to the true value. **Systematic error** or **inaccuracy** is quantified by the average difference (bias) between a set of measurements obtained with the test method with a reference value or values obtained with a reference method.

Precision refers to how well measurements agree with each other in multiple tests. **Random error** or **imprecision** is usually quantified by calculating the coefficient of variation from the results of a set of duplicate measurements" [58]. Difference between precision and accuracy can be seen in Figure 13.

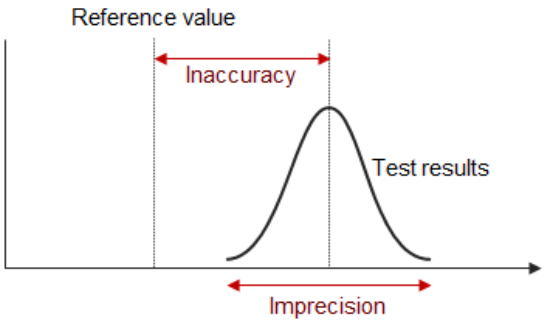

Figure 13. Difference between precision and accuracy [58]

Based on the definition of precision and accuracy, the **geodetic authenticity** can be defined in the following way. A GPS coordinate is defined to be geodetically authentic if it is precise (the calculation can be repeated with almost the same results) and accurate (the calculated position is the one, where the receiver is in the time of the positioning).

Generally, there are two main types of the weaknesses of the GPS, the one which is occurring because of the technical shortcomings or failures of the system, and the other which is caused intentionally and maliciously by an attacker. The table below lists the most critical vulnerabilities of GPS.

Table. 6. Weaknesses of GPS respect to geodesic authenticity

Vulnerabilities caused by a technical problem	**Threats caused by a malicious attacker**
Onboard atomic clock drift	Jamming
Satellite orbit change	Spoofing
Atmospheric effect (ionospheric/tropospheric delay)	
Multipath	
Interference	

Source of vulnerabilities occurred by technical problems can be found in the space, or the user segment of the GNSS/GPS. Nevertheless, most of the errors listed above induced by the space segment, such as the tiny drift of the extremely accurate atomic clocks on the satellites, where even a ten ns difference in the time can occur three meters positioning error on the ground. Alternatively, the subtle changes in the orbits of the satellites with incorrect ephemeris can cause two and a half meters difference in the position too. Atmospheric effects like ionospheric and tropospheric delay caused by the varying amount of electrically charged particles, and the never permanent humidity, pressure, and temperature, results in around 5 m error in the calculated position. Multipath and interference problems are caused by the wrong placement of the ground antennas and other radio sources transmitting signals because near the surface of Earth the GPS signal is extremely weak.

Threats made by intentional and malicious attackers are **jamming**, **spoofing** and meaconing as a subtype of spoofing**.** In some cases, meaconing is mentioned as a distinct attack, but most of the literature considers it as a subtype of a spoofing attack. In this dissertation, meaconing is considered as a subtype of spoofing too. These attacks result in inaccuracy or even total failure in the determined position, which therefore can threaten all the users of the system, like civil, military and critical infrastructures.

Jamming

According to Dovis [59], jamming is “blocking reception of the GNSS signal by deliberately emitting electromagnetic radiation (i.e., radio-frequency interference) to disrupt user receivers by reducing the signal-to-noise level.” So, the scenario is the following: the attacker tries to interrupt the connection between GPS satellites and GPS receivers, aiming the service entirely inaccessible, i.e. lost the connection or just disrupted it. In the case of jamming the aim is to prevent successful localization, navigation, and timing. It is somewhat an easy task, considering that the GPS satellites are orbiting in space 20.000 km far from Earth, and the signals are really weak when arriving the surface of Earth and the communication is happening over a wireless connection via radio. These weak signals are sensitive to the radio frequency

interference (RFI). Based on [60], [61], [62] jamming attack can be classified by two aspects; first is the cause of the attack, second is the used methods and technology. Jamming can occur because of two fundamental causes; the first one is a malicious attack; the second is protecting someone's privacy. In order to achieve any of the abovementioned goals, the attacker has to produce intentionally or accidentally an obstruction into the connection. The obstruction is usually some kind of device which can cause RFI. It can be done intentionally by GPS jamming gadgets, like "Personal Privacy Devices", or bigger transmitters - which induce radio frequency noise with a much wider working range or accidentally by using reserved radio frequencies or frequencies which are too close to the ones used by GPS. A wide range of hardware and software components is available to jam the GPS signals even though in many countries it is forbidden to use GPS jammers like in the USA or Hungary. Assuming the second classification aspect, jamming attacks are divided into two main groups too, **blanket jamming** and **deception jamming**. In the case of blanket jamming, the jammer generates such a strong radio signal resulting in a complete loss of GPS signals in receivers. Speaking of deception jamming the jammer broadcasts similar signals like GPS satellites transmit, but changing some critical parts of it to cause error decoding or error positioning. Both blanket and deception jamming attacks can be divided into subcategories; interested readers can look for more detailed information [60] and [62], while Table 7.also gives some insight.

Table. 7. Classification of jamming attacks

Cause of the attack	
	Harm someone maliciously
	Harm someone unintendingly
Technology	
	Blanket jamming
	Deception jamming

Spoofing

Spoofing is "transmission of counterfeit GNSS-like signals, with the intent to produce a false position within the victim receiver without disrupting GNSS operations" and "rebroadcasting of delayed GNSS signals without any distinction between the signal in space from different satellites" based on [59]. So, in contrast with jamming, at spoofing the connection is working correctly between satellites and receivers, sometimes a tiny gap or delay can be encountered only, so the users hardly or never detect that something went wrong. Spoofing can cause more dangerous or serious situations, because as the former president of the United Kingdom's Royal Institute of Navigation said: "Jamming just causes the receiver to die, spoofing causes the receiver to lie" [63]. The situation is usually the following: the attacker transmits a more powerful signal than the signal broadcasted by the GNSS satellites, or in the receiver falsifies the signal without even transmit a modified one. To be known that

an attack is in progress can be detected with different levels of success in the previous cases. If the attack is successful, then, the receiver will use the maliciously modified signal as it was the original one coming from the satellites without noticing the falsification. Spoofing attacks can be classified by several aspects, based on the comprehensive studies of van der Merwe et al. [64] and Jafarnia-Jahromi et al. [65]. The primary grouping property is the need or lack of signal transmission during the attack**. Radiofrequency (RF) based** / **GPS signal simulators** and **interface/receiver based** spoofing attacks can be distinguished. RF-based attacks using equipment, which transmits a falsified or an earlier recorded signal. Interface based attacks are using a module installed in the receiver, which modifies the incoming GPS signal or bypasses a modified signal. RF-based attacks can be classified by the number of used transmitters during the attack; single or multiple modes are available. Alternatively, by the method how the GPS signal simulators take over the place of the original GPS satellites in the communication, in this case, asynchronous or hard-take-over and synchronous or soft-take-over modes are distinguished. In the hard-take-over mode is not necessary to be aware of any information about the target. The attacker just generates a strong signal and suppress the original GPS signal with that stronger one. The hard-take-over attack is a brute-force attack. In soft-take-over mode, the attacker must have some information about the location of the target and transmits a signal with a continuously increasing power. The overlapping correlation peak and the increasing signal power makes the detection of this attack harder.

Furthermore, the aim of the spoofing can be served as a classification aspect; the aim can be the altering of the calculated position, time or other information – like any part of the NAV message – broadcasted by the GPS satellites. On the contrary, the interface-based attacks are altering or overwriting the already calculated position, velocity and time information with the help of a GPS module emulator inserted in the hardware level of the receiver, or malicious software installed on a higher level of the receiver. Table 8. displays the different types of spoofing attacks.

Table. 8. Classification of jamming attacks

Radiofrequency based - GPS signal simulator		
	Number of transmitters	Single Multiple
	Take-over method	Asynchronous Hard-take-over Synchronous Soft-take-over
	Aim	Position altering Timing and information altering
Interface or receiver-based	*Application*	GPS bypass Interface spoof

3.2.1.2 *WEAKNESSES OF GPS RELATED TO CRYPTOGRAPHIC PROPERTIES*

Only some information was available at the time of my research in the related scientific literature about the cryptographic reliability of the entire GNSS/GPS. On the other hand, this approach is an essential part of the system, because several GNSS based service exsists which has to provide cryptographically authenticated data – such as the Noise App, myGeoTracking, Alibi applications mentioned in Section 3.1.2 – so it would be essential to make location determination cryptographically secure and reliable. Communication always requires a channel on which the information can flow, speaking about GNSS the information is carried across space using radio waves. In Figure 14. the Shannon-Weaver communication model (cf. Section 2) can be seen with some extended information related to the communication in GPS.

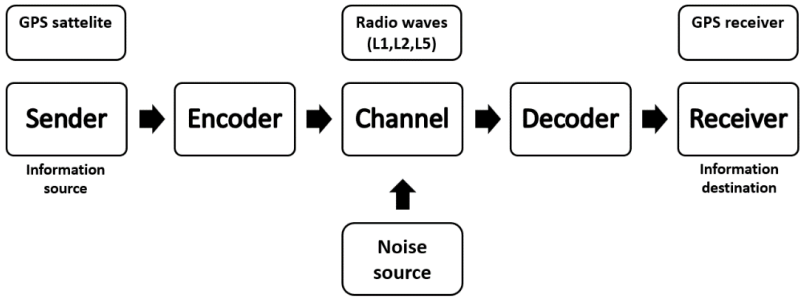

Figure 14. Extended Shannon-Weaver model of communication

The concept of the sender and receiver is used in the means of the Shannon-Weaver communication model. In the case of the communication between Space and User Segment of GPS, the information flow is one way, and there is no cryptographic protection installed on the channel at all. To be able to provide reliable information exchange, a secure communication channel is necessary, although such a channel will not guarantee, that the messages are ever received [10]. As it is explained in Section 2. a communication channel is considered secure if it provides the following properties between the GPS satellite and the GPS receiver during the transmission of the messages: confidentiality, authenticity, data integrity and freshness. Based on the related scientific literature the main problems of the GPS if we consider the space segment and user segment connection, is the lack of critical cryptographic properties, thus no secure channel can be provided, so confidentiality, authenticity, data integrity and freshness are not guaranteed parameters of the system. The following table summarizes what the following figure shows, namely the security breaches of the satellite-receiver communication of GPS.

Table. 9. Cryptographic vulnerabilities

Cryptographic vulnerabilities
Lack of confidentiality
Lack of authenticity
Lack of data integrity
Lack of freshness

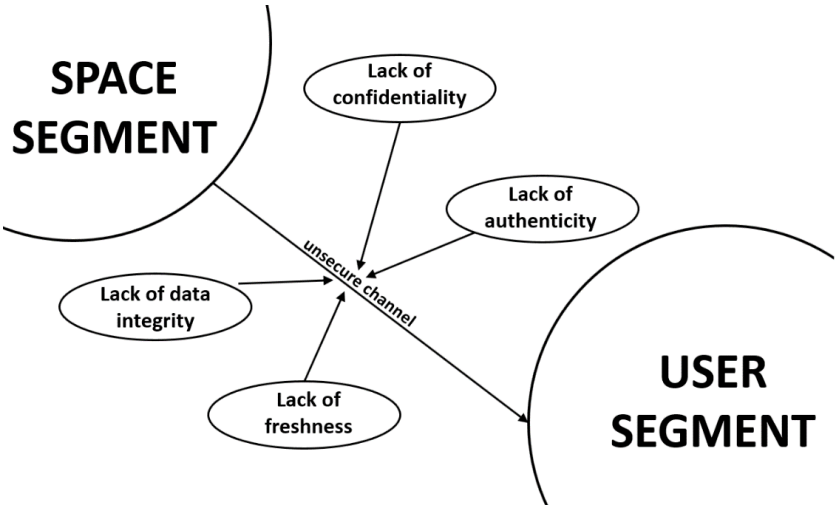

Figure 15. Security breaches of GPS

Communication between the user segment and the space segment in the GPS is one way, as it is mentioned before, the satellites cannot receive any message from the users, only transmit information predefined and refreshed by the control segment, and receivers are only able to listen to the signals transmitted by the satellites. An important consequence of that property of the communication is that the receiver cannot check the identity of the satellites and this is true in the other direction as well, so the authentication of any participant is not provided in the system, because none of the participants is in the knowledge of the identity of the other party. Lack of entity and message authentication in the GPS can threaten all the users of the system, like civil, military and or critical infrastructures. Speaking of the SPS the messages transmitted over radio waves without any encryption method applied. It means that anybody can access the content of the messages; only a GPS receiver is necessary. The lack of the encryption of the messages resulted that confidentiality is not provided by the system either, because the content of the messages is not hidden from the unauthorized users. Although in the case of legacy NAV Hamming codes are applied to detect errors in the message, and for modern LNAV Cyclic Redundancy Check codes are used, no method is available to check if the content of

the messages is the same at the time of the dispatch and the arrival. So, data integrity is not provided either, because it cannot be checked that the messages are modified during the communication or not.

3.2.2 EXISTING SOLUTIONS FOR THE REVEALED WEAKNESSES

3.2.2.1 SOLUTIONS FOR THE GEODETIC PROBLEMS

Geodetic weaknesses like jamming and spoofing are exhaustively examined in the related literature, and several solutions are performed. Reviews of **antijamming** [66], [67] and **antispoofing** [65] techniques are overviewed in this section.

Speaking of spoofing, two main types of antispoofing methods are distinguished, the **spoofing detection** and the **spoofing mitigation**. In the case of spoofing detection, the provided solution only helps to discriminate against the spoofing signal but does not do anything to eliminate it. Several spoofing detection methods are available, in this section the methods collected by Jafarnia-Jahromi et al. [65] are examined, Table 10. contains a non-completed list about existing spoofing detection methods, in this section a part of those methods s explained, a detailed explanation can be found in [65].

Table. 10. Spoofing detection methods

Type of spoofing detection	Method
Signal Power Monitoring	C/N_0 Monitoring
	Absolute Power Monitoring
	Received Power Variations versus Receiver Movement
	L1/L2 Power Level Comparison
Spoofing Discrimination Using Spatial Processing	Multiantenna Spoofing Discrimination
	Synthetic Array Spoofing Discrimination
Time of Arrival Discrimination	PRN Code and Data Bit Latency
	L1/L2 Signal Relative Delay
Consistency Check with other Navigation and positioning Technologies	

The most obvious solution is to monitor the intensity of the received signal, or some parameter of it, because if the spoofer is not sophisticated enough, then the spoofing signal is much stronger than the GPS signal, methods like this called **Signal Power Monitoring**. The first method of this kind is the $\boldsymbol{C/N_0}$ **(carrier to noise density) Monitoring**, where the value of the C/N_0 is observed. C/N_0 is "an indication of the signal power of the tracked satellite and the noise density as seen by the receiver's front-end" [68]. A sudden change – in that particular parameter of the system – can quickly reveal the presence of a spoofing signal. **Absolute Power Monitoring**

method is measuring the power of the received signal in a general manner. If the amplitude of the received signal is higher than usual, the spoofer is revealed. For constantly moving GPS receivers the C/N_0 can be observed too, because if the system is working well, this parameter does not change during movement. But if the spoofer uses a single fixed antenna, then the increasing or decreasing distance between that antenna and the target receiver will cause a continuously varying value of C/N_0 , which may indicate that a spoofing signal is received. The method explained before named **Received Power Variations versus Receiver Movement** method. If the receiver is able to listen to multiple frequencies in parallel, e.g. L1 and L2, then the **L1/L2 Power Level Comparison** can be used. Which is with checking the phase difference between the two carrier frequencies can detect a spoofer, because that difference value is well-known and constant.

It often happens that an antenna is broadcasting more than one spoofing signal, so the origin of these signals is the same, unlike in the case of GPS signal, where the signal is coming from different satellites from different directions. **Spatial processing method**s are taking advantage of the previous limitations. Two types of spatial processing techniques are mentioned in [65], the **Multiantenna Spoofing Discrimination** and the **Synthetic Array Spoofing Discrimination**. In the case of Multiantenna Spoofing Discrimination, two fixed antennas are used in a single receiver. A theoretical phase difference is calculated between the two antennas, and then during the processing of GPS signals, a practical phase difference is calculated continuously. If a spoofing signal is present, then the theoretical difference and the practical difference will be completely different. Synthetic Array Spoofing method does the same, without using multiple antennas. A single GPS receiver is moving around a random path, and the phase of the received signal is calculated at some random points. If the phase values – calculated at those different points – are not correlated, then a spoofing signal can be guessed.

Time of Arrival Discrimination technics are available too, in the **PRN Code and Data Bit Latency** method a delay in the received signal is watched, if it is greater than 20 ms, then a spoofer signal can be guessed. This discrimination method provides a solution against receiver-based spoofers, where the attacker first receives the original GPS signal, and only sends it towards after modifying it. Receivers which are capable of receiving more than one signal at the same time can use the **L1/L2 Signal Relative Delay** method, which is based on the fact, that because of ionospheric and tropospheric effect there is a tiny delay in the transmission time of the P(Y) code on both of the carrier signals mentioned above.

The **Consistency Check with other Navigation and Positioning** technologies are using the help of a cellular or Wi-Fi network to monitor the proper working of the

GPS receiver. However, geodetic authenticity of location information provided by Wi-Fi and cellular networks always has a lower degree than the GPS. Differences between the location information provided by the GPS and the Wi-Fi or cellular network can reveal a spoofing signal.

In the case of signal mitigation, the main aim is to eliminate the spoofing signals. There are technics – namely the **Vestigial Signal Detection** methods – which try to find some trace of the original GPS signal. One solution is to record the incoming data into a buffer, then remove that part of the data, which is coming from a particular GPS satellite. After deleting that part from the buffer, start a new search for the same GPS satellite signal and what remained is the original unmodified signal. Regarding to **Multiantenna Beam Forming and Null Steering** methods, a spatial spoofing detection method is applied to find the direction of the spoofing signal, and then with a radio wave, which amplitude and frequency is the same but antiphased cancel out the spoofing signal in that particular receiver. Sometimes it is easier to observe the output of the location determination than the received signal. The **Receiver Autonomous Integrity Monitoring (RAIM)** method is applying an integrity checking – which is based on statistical methods – in the measurement of the receiver, during the calculation to remove the wrong location information.

Table. 11. Spoofing detection methods

Spoofing mitigation method
Vestigial Signal Detection
Multiantenna Beam Forming and Null Steering
Receiver Autonomous Integrity Monitoring

Regarding to jamming, it is only necessary to pay attention to the jamming mitigation, because detection of the jamming attack is easy, if the signal is lost, then a jamming attack is in progress. Mitigation methods listed above in the case of spoofing can be used to mitigate the jamming attacks as well but based on the open literature, the most commonly applied technologies are the anti-jam antennas. Anti-jam antennas have the advantage that no changes have to be made neither in the GPS service nor in the receiver. Anti-jam antennas are easily applicable antenna enhancements for the GPS receivers. Anti-jam antennas are **Controlled Reception Pattern Antennas (CRPA)** [69] which using spatial methods to identify the direction of the broadcasted jamming signal and then eliminate the interference with generating a phase-destructive sum of the interference [70].

3.2.2.2 SOLUTIONS FOR THE CRYPTOGRAPHIC PROBLEMS

At the beginning of the 2000s, because of the widespread usage of GPS, several researchers started to design solutions for the authentication shortcomings of the

system, mentioned above. A lot of possible methods were made through the early years of the 2000s. Cryptographical GPS signal authentication came in mind regarding spoofing and meaconing attack as an alternative countermeasure. Although a proper cryptographically authenticated signal can provide a solution for legal problems in the case of applications presented in Section 3.1.2 too. As it is mentioned before in this dissertation, the focus is on those types of applications where the received GPS coordinates have probative value in a legal procedure or dispute.

It is well known that electronically saved, and transmitted data without a secure channel can be modified easily. Hence, it is necessary to have a method which is able to prove that the origin and the destination (authenticity), the content (data integrity), and the freshness of data provided by the GPS is not altered. Thus, in this section, the related scientific literature is overviewed to get an insight into the already existing cryptographical methods for authenticating GPS signals.

In cryptography when a communication channel has to be secured, the following methods are available: symmetric or asymmetric (or public) key algorithms and cryptographical hash functions (from now on hash functions). In Section 2, a brief explanation can be found about the cryptographic algorithms mentioned above, for more detailed information see [12] and [17].

In symmetric-key cryptography, as it is mentioned in Section 2, a shared secret key is used between the two parties of the communication, and with this common key, the messages can be encrypted and decrypted on a channel. If the shared key cannot be revealed, then the confidentiality and the data integrity properties of the secure channel are provided. However, these methods cannot assure the authenticity of the messages or the entities and the freshness of the messages. Another problem is that considering the fact that millions or billions of GPS receivers are used all around the world, and every GPS satellite should share a secret key with all those receivers and should use that particular secret key in that particular communication. Thus, it can be stated that symmetric key cryptography is not the right choice for GPS signal authentication.

In the case of asymmetric key algorithms, as it is described in Section 2 a pair of a secret/private and a public key is used to ensure the confidentiality, the data integrity, and the entity authentication during the communication. Only the private key must be kept secret, and there is no need for key exchange via a secure channel. The digital signature algorithm is used in existing solutions. In the case of digital signature, the sender entity – the satellite –, that has to be authenticated in the communication using its private key signs the message, and on the other side the receiver entity using the public key – paired to the private key of the to be

authenticated sender entity – verifies the message. Although digital signature requires less effort when the secrecy of the keys is under observation, but the underlying mathematical methods and its implementation take much more time. Modern GNSS, like GALILEO, uses an asymmetric cryptographical method to digitally sign the NAV messages [71].

Hash functions as the third option, are one-way functions with an arbitrary length input and a fixed-length output. Hash functions are in possession of the properties such, the output is easy to compute for any given input, it gives the same output for one particular input (deterministic), the output is not revealing any information about the input (preimage resistance), and it is practically impossible to find two different inputs to get the same output (collision resistance). Keyed hash functions are able to provide not just data integrity but also message authentication; unkeyed hash functions provide message authentication beside message integrity, that is the reason why it serves as a basis of several solutions.

Based on [59] and [72], the cryptographic signal authentication methods can be divided into four categories, like **Navigation Message Authentication (NMA)**, **Spreading Code Authentication (SCA)**, **Navigation Message Encryption (NME)**, and **Spreading Code Encryption (SCE)** as it can be seen in Figure 16.

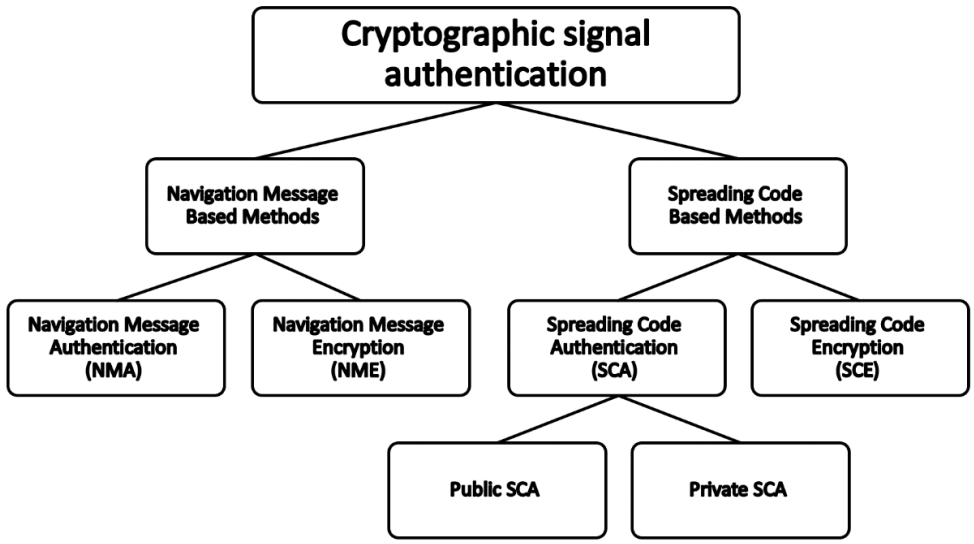

Figure 16. Classification of cryptographic signal authentication

NMA "denotes the authentication of satellite navigation message by means of digitally signing the navigation message data and thus keeping the navigation message clear (i.e. unencrypted)" [59]. The aforementioned statement is true only in the case of NMA designed for GPS, because for example, in the case of GALILEO,

the NMA is based on a symmetric cryptographic algorithm. In the case of NMA technics, the whole NAV/CNAV/CNAV2 or some part of it is digitally signed by the GPS satellites. All satellites have a private-public key pair to be able to perform the digital signing, which as it is mentioned before an asymmetric cryptographic algorithm. The digital signature of the data is broadcasted standalone or in the message itself. Legacy GPS does not have a suitable structure to allow the signature encapsulation into the message, but modern navigational messages have additional places in its frame to bear such extra information. After receiving the original navigational message and the digitally signed version of it, the receiver is able to verify the validity of the signature. The main disadvantage of the NMA that a delay is occurring during the authentication, because the receiver has to wait until the whole navigational message and the signed message arrives, before the verification. So, the NMA does not authenticate the entity, only the actual message. There can be presented several solutions according to what amount of data is signed digitally, or the applied digital signature method, the key generation, and the process of the key publishing. Another disadvantage of the NMA that a major change is required in the structure of the signal and the receivers as well. NMA first was proposed by S. Logan in [73], as a basic level authentication technic for L2C and L5 GPS signals. In [74], [75], [76], [77], [78], other possible solutions are proposed for future NMA.

SCA "embeds short encryption code segments within the nominal (unencrypted) spreading code sequence" [59]. The SCA method is first mentioned by Logan in [73] as an intermediate level of an authentication method for GPS signals. The SCA extends the NMA method by inserting extra parts into the spreading code, so into the C/A and Y code itself, before it is modulated onto the carrier frequency. The code period has become longer because of the extra chips. The SCA takes advantage of the low transmission power of the spreading code because it is transmitted 20 dB below the background noise level, so ordinary receivers cannot observe those codes. The receivers are only able to cross-correlate a known spreading code sequence with the transmitted ones. "The SCA chips are therefore obscured by the thermal noise in a similar way as a message written by invisible ink" [72]. Encrypted spread sequences have to be known by the receivers to be able to determine the position. Several methods are existing, to determine the spreading code sequence. One is when the spreading code sequences are extended by a secret key, which then used to encrypt the transmitted navigational message, mentioned in [79]. In that solution, it is not necessary to store the secret key in the receiver side, only a seed value and the key generating method are the information what has to be known. Another solution is when the unsent digitally signed navigational message is served as a seed for the spreading code sequence, proposed in [73]. Both methods have the advantages of what NMA methods are provided and the extra benefits provided by the SCA.

Methods like this usually called hybrid systems, in the related scientific literature, several methods are published like Chimera in [79], or Private SCA and Public SCA in [73]. The new authenticated signal of the European GNSS Galileo is protected by an NMA SCA hybrid system presented in [80].

NME "refers to the encryption of the whole navigation message, which is then modulated on the spreading code" [59]. In the case of NME, usually, symmetric cryptographic algorithms are in use, so a single secret key is helping to encrypt and decrypt the whole navigation message. It is explained above, that symmetric cryptographic algorithms have several disadvantages in usage for signal authentication purposes, mainly in the case of civil GPS, because of the vast number of users. The distribution of the secret key can be hard to realize, and tamper-resistant devices have to be applied, to keep the key in secret. Any type of receiver is able to receive the broadcasted encrypted navigational message, but only the authorized ones are able to understand the content of it. In the related scientific literature, only a few proposals for NME can be found, for example, the one in [77].

SCE "denotes the encryption of the whole spreading code sequence transmitted by each satellite" [59]. In the case of SCE symmetric cryptographic algorithms are used like NME, especially stream ciphers. SCE is most commonly used in military applications, where the number of users is significantly less. The military P(Y) code is secured with SCE technics. P-code encryption and Y code decryption are solved by using three different secret keys, the **Group Unique Variable** (**GUV**), the **Cryptovariable Weekly** (**CVw**) and the **Cryptovariable Daily** (**CVd**). CVd is the key that is used to encrypt the P-code to get Y-code, so every satellite has the CVd key. Y code and encrypted CVd (with GUV) are transmitted by the satellites. To decrypt the Y code and get the P-code, the receivers need the CVd key. If the receiver is in possession of the annual GUV, then with the help of it the CVd can be decrypted from the transmitted message although this solution causes a delay in the position determination because the whole Y code and the encrypted CVd have to arrive before decryption. The other method provides a faster position determination. If the receiver has the CVw, then it can generate the actual CVd, like the satellite, and does not have to wait till the whole encrypted section arrives. SCE methods are not common, and it is usually mentioned in the literature in connection with GPS P(Y) code [77] , [81].

It is essential to mention that all the above presented solutions require fundamental changes in the structure of the GPS signals, so without the evolution of the system they cannot be applied.

3.2.3 OWN SOLUTIONS: GENERATING CRYPTOGRAPHICALLY AUTHENTIC LOCATION INFORMATION

3.2.3.1 DESCRIPTION OF OUR PROTOCOLS

In order to solve the problems formulated in Section 3.2.1, and provide cryptographically authentic location information, two stamping protocols were made by my co-author and me in 2013 [82]. The developed protocols can provide authentic location and time information for any device which has a GPS receiver built in it. In this section, both protocols are presented, one with a higher safety solution, which was designed to be installed at the hardware-level of the host devices, and the other one – that provides lower safety – was designed to be installed at the software-level of the host devices.

During the design of communication protocols, several rules have to be applied while the desired properties of the protocol are considered. The primary aim of the design process to eliminate all the known security breaches/attacks as far as possible. In the case of cryptographic protocols, it is even more important to be cautious, because it is not enough to pay attention to the proper working of the underlying transmission channel – which is essential –, like that the channel is available, stable, and there is no loss of information though it. Nevertheless, such a protocol has to provide a secure channel with its special properties as it is defined in Section 2., namely confidentiality, authenticity, data integrity, and freshness. To achieve a protocol with an appropriate safety level, proper cryptographic primitives have to be used as building blocks during designing. However, using the right primitives will not ensure automatically the same level of security for the protocol.

Before, the two protocols are explained in detail, the starting point of our research is presented. In 2001 Alf Zugenmaier and Matthias Kabatnik [83] introduced a „location-stamp" service for mobile telephone networks. The solution of theirs provides authenticated cell information, calculated previously by the method of trilateration from three different cell towers. They "propose a service that provides certified location information. Integrated with cryptographic digital signatures, this service enables the determination of the current position of the signer in a provable way. The „location-stamp" service (LSS) provides a „location-stamp" that certifies at which position the subscriber is located when making a signature—or rather, at which position the subscriber's mobile station is visible to the network. This „location-stamp" can be verified by any third party using the public key assigned to the LSS, and it can be used as a proof of the location information" [83]. Their protocol served as a basis of our solutions. However, the method presented by them is impossible in its original form to use in the case of authenticating GPS coordinates, because of the two-way communication between the trusted third party – for

example, the Certification Authority – and the Location Measurement System – for example a mobile service provider –, when cell information authenticated. In our case, the Location Measurement System is the GPS, not the cell towers of any mobile service provider, where the communication is one-way. Because the GPS satellites – as it is thoroughly explained in Section 3.1.1 – only can send the information but cannot receive it.

Here two solutions for the cryptographic authentication of the GPS coordinates are suggested. The basic idea is that the data received and/or computed by a GPS receiver, or by a mobile phone, tablet, laptop, smartwatch along with others that has a GPS receiver built-in it (from now on device), are sent to a trusted organization, which can be for example a Certification Authority. The device has to use digital signing, and if the information is consistent with information available by the organization, then the organization confirms the validity of the GPS coordinates with signing it digitally.

The basic difference between the two solutions is that in the first case it is assumed that the navigational message and additional information (from now on raw data) – explained in Section 3.1.1 – transmitted by the satellites and received by the device – is accessible; thus it can be signed. Differently, in the second case, only the computed GPS coordinates are accessible. So, the first protocol is more promising, although the safety level of the second one can be satisfactory in the case of most of the applications.

Working with the GPS requires processing a large amount of data, such as the raw data arriving from the satellites, time information or the calculated coordinates. The task of cryptography is to prevent these data from changing during the process of the calculations. The changes can be made, for example by a malicious person, a virus, a modified receiver, or modified software and sometimes it may happen by chance. As for the precision of the device, there are several options for the reliability:

- First, the person and the device are both considered trusted in all cases. The measured coordinates and the time provided by the device are accepted.
- Second, the person and the device are both considered trusted, but the time information of the device is not acceptable. The measured coordinates are accepted, but the time provided by the device is not.
- Third, the person and the receiver are both considered unreliable. Neither the measured coordinates nor the time provided by the device is accepted.

In this thesis, the third option is considered, and two solutions are suggested to solve it. The first solution provides higher safety, with the creation of an authenticated „location-stamp” based on the hardware-level of the device. The second solution is

using the data preprocessed by the software of the Mobile Device to generate the authenticated „location-stamp”, and the second method provides lower safety. The differences between the abovementioned two solutions are the following: In the first case, the Authentic Software is built in the hardware level of a Mobile Device. The data are signed immediately as the device received it. Thus, the provided security is hardware dependent. In the second case, the Authentic Software is on the level of the operating system, so the provided security is software dependent. Thus, the second solution security is improved with trilateration for the Mobile Device, between the cell towers.

The following cryptographic primitives are used during the operation of the protocols:

- digital signature
- hash function
- timestamp

3.2.3.2 High-safety solution: Hardware-level

In this solution, the main goal is to take the raw data before anybody could modify it. The Authentic Software developed by us is intended to build in a very deep and hidden layer of the device; thus, it is built in the hardware level of the Mobile Device. Figure 17. displays the protocol.

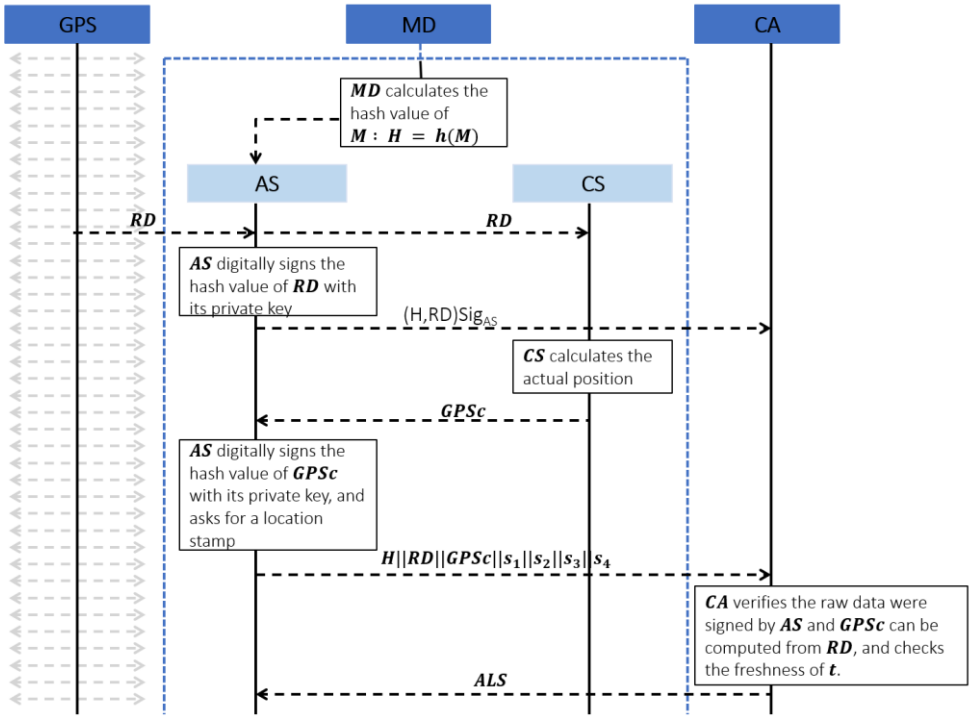

Figure 17. Protocol for hardware- level solution

Table. 12. Participants and Notations

Denotation	Explanation
$\boldsymbol{GPS}$	The Global Positioning System. The satellites of this system provide the data from which the GPS receiver calculates the coordinates of the actual position.
$\boldsymbol{MD}$	The Mobile Device. The device with a GPS receiver, with the help of which the positioning and authentication can be made.
$\boldsymbol{CA}$	The Certification Authority. It provides the authentic time and „location-stamp"; this is an organization, which is independent of the measurement and can guarantee that nobody modified the results.
$\boldsymbol{AS}$	Authentic Software. This software makes the authentication of the raw data (which comes from the satellites) and of the calculated GPS coordinates.
$\boldsymbol{CS}$	The Calculator Software. This software calculates the GPS coordinates belonging to the actual position from the raw data come from the satellites.
$\boldsymbol{M}$	The text or photo or some other data that we want to authenticate with a „location-stamp".
$\boldsymbol{h}(\dots)$	The hash function.
$\boldsymbol{H}$	The hash value of M.
$\boldsymbol{c}(\dots)$	The calculator function, which calculates the current position from the raw data that come from the satellites.
$\boldsymbol{RD}$	The raw data come from one of the satellites of the Global Positioning System.
$\boldsymbol{GPSc}$	The GPS coordinate calculated by the Calculator Software.
$\boldsymbol{S_{AS}}(\dots)$	The signature of the data in parentheses with the private key of the Authentic Software.
$\boldsymbol{S_{CA}}(\dots)$	The signature of the data in parentheses with the private key of the Certification Authority.
$\boldsymbol{V_{AS}}(\dots)$	The verification of the data in parentheses with the public key of the Authentic Software.
$\boldsymbol{V_{CA}}(\dots)$	The verification of the data in parentheses with the public key of the Certification Authority.
$\boldsymbol{s_i}$	The i-th signed data.
$\boldsymbol{TIME}$	The time information.
$\boldsymbol{n}$	The nonce value
$\boldsymbol{ALS}$	The authentic „location-stamp" generated by the Certification Authority.

Below the detailed steps of the higher-safety protocol can be seen.

1. MD calculates the hash value of
   $\boldsymbol{M: H = h(M)}$
2. $\boldsymbol{MD \rightarrow AS: H||M}$
3. AS digitally signs H with its private key:
   $\boldsymbol{s_1 = S_{AS}(H)}$
4. $\boldsymbol{GPS \rightarrow AS: RD}$
5. AS digitally signs the hash value of RD with its private key:
   $\boldsymbol{s_2 = S_{AS}(h(RD))}$
6. $\boldsymbol{AS \rightarrow CS: RD}$
7. CS calculates the actual position from
   $\boldsymbol{RD: GPSc = c(RD)}$
8. $\boldsymbol{AS \leftarrow CS: GPSc}$
9. AS digitally signs the hash value of $GPSc$ with its private key:
   $\boldsymbol{s_3 = S_{AS}(h(GPSc))}$
10. AS concatenates H, s_1, RD, s_2, $GPSc$ and s_3 and takes its hash value and then digitally signs this hash value with its private key:
    $\boldsymbol{s_4 = S_{AS}\,(h(H||RD||GPSc||\,s_1||s_2||s_3))}$
11. $\boldsymbol{AS \rightarrow CA: H||RD||GPSc||s_1||s_2||s_3||s_4}$
12. CA verifies that the raw data were signed by AS and CA verifies that $GPSc$ can be computed from RD, and checks the freshness of t.
    $\boldsymbol{V_{AS}(s_2)}$
    $\boldsymbol{c(RD)\ =?\ GPSc}$
    $\boldsymbol{f(t)}$
    12.1. if the answer is true for all questions, then CA makes the authentic „location-stamp”:
    $\boldsymbol{ALS}$
    $\boldsymbol{= TIME\,||S_{CA}\,(h(H||RD||GPSc||n||s_1||s_2||s_3||s_4||s_5||s_6||TIME))}$
    $\boldsymbol{AS \leftarrow CA: ALS}$
    12.1.1. AS verifies that really the CA signed the „location-stamp” that it got:
    $\boldsymbol{V_{CA}\,(ALS)}$
    12.1.1.1. if the answer is true, then AS accepts the authentic „location-stamp”
    12.1.1.2. if the answer is false, then AS starts a new „location-stamp” request with step 4.
    12.2. if the answer is false, then CA rejects to generate the authentic „location-stamp”
    $\boldsymbol{AS \leftarrow CA: rejection}$

The protocol, described here in detail, has three important participants, these are the satellites of the Global Positioning System, the Mobile Device, and the Certification Authority. The Mobile Device generates a print of the data, – which has to be stamped with an authentic „location-stamp" – initially with an eligible hash function. Applying a hash function is necessary because of the digital signing. In the next step the Authentic Software, which is built in the hardware of the Mobile Device, gets the data from the three GPS satellites, and then it digitally signs these data with its private key immediately. Digitally signing the raw data, that is arriving from the satellites, is required in order to nobody falsify them during the computational process. The Authentic Software located in the hardware of the Mobile Device so that it can protect the data from the attack of any software/application installed on the operation system of the Mobile Device. Once the Authentic Software digitally signed and stored the raw data, it sends them to the Calculator Software. The Calculator Software calculates the current GPS coordinates from the actual raw data and sends back the result to the Authentic Software. The Authentic Software digitally signs these data too. Now we arrived at the point where the Authentic Software can ask – if it has all the information which is necessary – for an authentic time stamp from the Certification Authority. So, the Authentic Software sends a request to the Certification Authority. This request contains the hash value of all the data, the raw data and the calculated coordinates concatenated and digitally signed with its private key. Then the Certification Authority generates a nonce value in order to ensure the freshness of the protocol finally gives it back to the software. The Authentic Software appends the nonce to the previous request and turns it back to the Certification Authority, which checks if it is true that the Authentic Software sent the request. If the result of the verification is right, then the Certification Authority checks that the calculated GPS coordinates can be or cannot be computed from the raw data. If the answer is yes, then it generates the „location-stamp", which includes a time stamp too.

3.2.3.3 Lower-safety solution: Software-level

The protocol of the previous section strongly depends on the available hardware, because the raw data – received by the GPS device from the satellites – is signed immediately after the device got it and it can be done only on a very low level of the Mobile Device. A further aim with the second protocol is to develop a solution, that is less hardware dependent because it is much easier to access the calculated GPS coordinates as the raw data. In order to authenticate the GPS information of the Mobile Device, a trusted organization is necessary, that has its own data which can be compared to the ones received by the Mobile Device. One option to get suitable data is using mobile phone services, which have cell information, but cell information is usually not accurate enough to specify the location of the Mobile Device. It seems to be a proper solution if the mobile network coverage is broad enough. In the above situation, the mobile phone service has independent information on the location of the Mobile Device, which can be compared to the data that the Mobile Device sends to the trusted organization. The second protocol is only applicable if the above assumptions hold. In the next, the details of the protocol are presented. Table 13. contains the participants of the protocol and the important notations. Only those symbols are presented, which differs from participants or notations in the case of the previous protocol; other symbols are used in the same manner as above.

Table. 13. Participants and Notations

Denotation	Explanation
$\boldsymbol{MPSP}$	The Mobile Phone Service Provider provides the cell information for a mobile identifier.
$\boldsymbol{AS}$	Authentic Software. This software makes the authentication for the calculated GPS coordinates.
$\boldsymbol{MoID}$	The Mobil identifier, a number from which the MPSP can identify the current Mobile Device.
$\boldsymbol{CI}$	The cell information from the MPSP.
$\boldsymbol{t}(\dots)$	The trilateration function, it trilaterates the CI from the MoID.
$\boldsymbol{ck}(\dots)$	The checking function, which checks if GPS coordinates are in the area which is defined by the cell information.
$\boldsymbol{S_{MPSP}}(\dots)$	The signature of the data in parenthesis with the private key of the Mobile Phone Service Provider.
$\boldsymbol{V_{MPSP}}(\dots)$	The verification of the data in parenthesis with the public key of the Mobile Phone Service Provider.

Figure 18. displays the protocol.

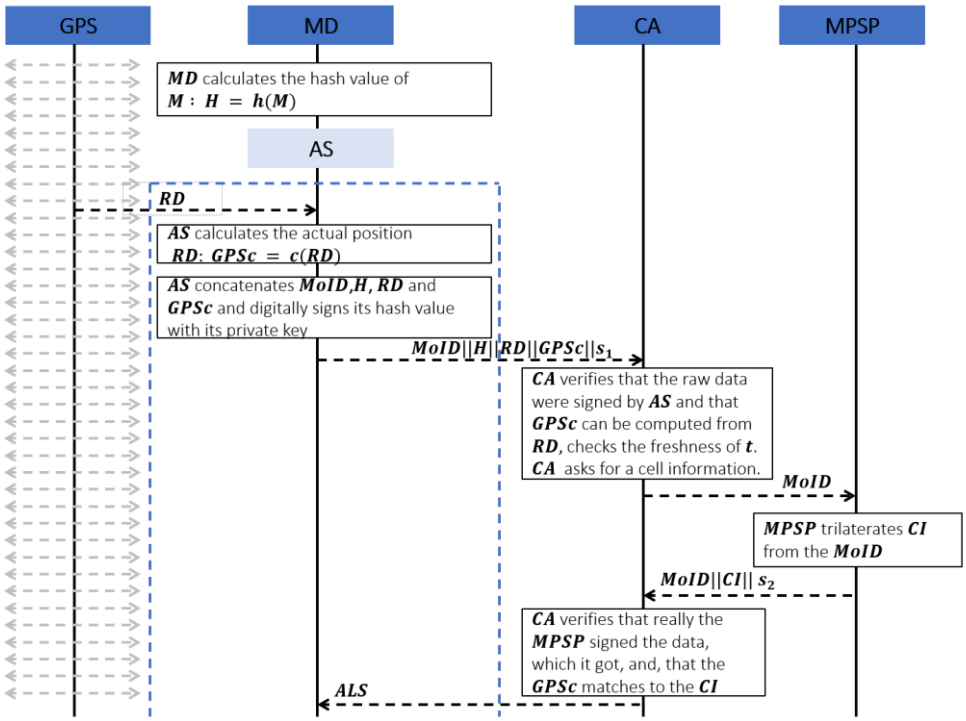

Figure 18. Protocol for software - level solution

Below the detailed steps of the software-safety protocol can be seen.

1. MD calculates the hash value of
   $\boldsymbol{M}: \boldsymbol{H} = \boldsymbol{h}(\boldsymbol{M})$
2. $\boldsymbol{MD} \rightarrow \boldsymbol{AS}: \boldsymbol{H}||\boldsymbol{M}$
3. $\boldsymbol{GPS} \rightarrow \boldsymbol{AS}: \boldsymbol{RD}$
4. AS calculates the actual position from
   $\boldsymbol{RD}: \boldsymbol{GPSc} = \boldsymbol{c}(\boldsymbol{RD})$
5. AS concatenates $MoID$, H, RD and $GPSc$ and digitally signs its hash value with its private key: $\boldsymbol{s_1} = \boldsymbol{SAS}(\boldsymbol{h}(\boldsymbol{MoID}||\boldsymbol{H}||\boldsymbol{RD}||\boldsymbol{GPSc}))$
6. $\boldsymbol{AS} \rightarrow \boldsymbol{CA}: \boldsymbol{MoID}||\boldsymbol{H}||\boldsymbol{RD}||\boldsymbol{GPSc}||\boldsymbol{s_1}$
7. CA verifies that the raw data were signed by AS and CA verifies that $GPSc$ can be computed from RD and checks the freshness of t.
   $\boldsymbol{V_{AS}}(\boldsymbol{s_1})$
   $\boldsymbol{c}(\boldsymbol{RD}) =? \boldsymbol{GPSc}$
   $\boldsymbol{f}(\boldsymbol{t})$
   7.1. if the answer is true then:
   $\boldsymbol{CA} \rightarrow \boldsymbol{MPSP}: \boldsymbol{MoID}$ and asks for a cell information
   7.1.1. $MPSP$ trilaterates CI from the $MoID$:
   $\boldsymbol{t}(\boldsymbol{MoID}) = \boldsymbol{CI}$
   $\boldsymbol{s_2} = \boldsymbol{S_{MPSP}}(\boldsymbol{MoID}||\boldsymbol{CI})$
   $\boldsymbol{CA} \leftarrow \boldsymbol{MPSP}: \boldsymbol{MoID}||\boldsymbol{CI}||\boldsymbol{s_2}$
   7.1.2. CA verifies that really the $MPSP$ signed the data, which it got, and, that the $GPSc$ matches to the CI:
   $\boldsymbol{V_{MPSP}}(\boldsymbol{s_2})$
   $\boldsymbol{ck}(\boldsymbol{GPSc}, \boldsymbol{CI}) = \boldsymbol{true}$
   7.1.2.1. if the answer is true, then CA makes the authentic „location-stamp”:
   $\boldsymbol{ALS} =$
   $\boldsymbol{TIME}||\boldsymbol{S_{CA}}(\boldsymbol{h}(\boldsymbol{MoID}||\boldsymbol{H}||\boldsymbol{RD}||\boldsymbol{GPSc}||\boldsymbol{s_1}||\boldsymbol{s_2}||\boldsymbol{CI}||\boldsymbol{TIME}))$
   $\boldsymbol{AS} \leftarrow \boldsymbol{CA}: \boldsymbol{ALS}$
   7.1.2.1.1. AS verifies that really the CA signed the „location-stamp” that it got:
   $\boldsymbol{V_{CA}}(\boldsymbol{ALS})$
   7.1.2.1.1.1. if the answer is true, then AS accepts the authentic „location-stamp”
   7.1.2.1.1.2. if the answer is false then AS starts a new „location-stamp” request with the step 3.
   7.1.2.2. if the answer it false then CA asks a new cell information with the step

7.2. if the answer is false, then CA rejects to generate the authentic „location-stamp"
$\boldsymbol{AS \leftarrow CA: rejection}$

Unlike the other method, described in Section 3.2.3.2, here the protocol has four important participants, the satellites of the Global Positioning System, the Mobile Device, the Certification Authority, and the Mobile Phone Service Provider of the actual Mobile Device. There are some other differences between the two protocols: in the case of the protocol presented in Section 3.2.3.2 the Authentic Software is built in the hardware level of the Mobile Device, here the software is installed in the operation system of the Mobile Device. The Mobile Device initially generates a print of the data with a hash function the same way as it happens in the previous case. After this, the Authentic Software, which is installed in the Mobile Device, gets the data from three GPS satellites and calculates the current GPS coordinates incoming from the raw data. Then it concatenates these two values, the document hash value, and the Mobile Device identifier and digitally signs it with its private key. As a next step, the Authentic Software asks for an authentic time stamp from the Certification Authority with the help of the digitally signed data. The Certification Authority generates a nonce value in order to ensure the freshness of the protocol and sends it back to the software. The Authentic Software appends the nonce to the previous request and sends it back to the Certification Authority, which checks that the Authentic Software sent the request. Now the Certification Authority sends the identifier of the Mobile Device to the Mobile Phone Service Provider, which trilaterates the cell information for the device and gives it back. At this point, only the verification remains behind. If the result of the verification is right, namely the calculated GPS coordinates match to the cell information, and the private key belongs to the authentication software. The Certification Authority generates the „location-stamp", which also includes a timestamp too.

3.2.3.4 SECURITY ANALYSIS OF OUR PROTOCOLS

Proving that a cryptographic communication protocol is working as it is promised is crucial. To be able to check if a security property is holding in a communication protocol, first, the protocol has to be modelled mathematically. In this section, the comprehensive survey of Blanchet [84] is overviewed, and I publish the content of the section in [85]. Several methods are proposed in the related scientific literature, and two main approaches of modelling are distinguished the **symbolic** (Dolev-Yao) and the **computational** one. Dolev-Yao model is highly abstract, while the computational model is more realistic. Dolev-Yao model represents cryptographic primitives by function symbols – it cannot be known, what is happening in the primitive –, and the messages in a communication flow are terms of the beforementioned function symbols. The attacker is able to compute with the defined

cryptographic primitives only. On the contrary, in the computational model, the cryptographic primitives are represented with the function of bitstrings, where the messages are the bitstrings themselves. The attacker is a probabilistic Turing machine, so the next step of the attacker is randomly restricted to some probabilistic value. Models such as the Dolev-Yao or the computational are made with the promise to prove security properties, like confidentiality, authenticity, data integrity or freshness provided by such protocols described above. To simplify the verification, automatic verification tools are designed for both models. The verification methods can be classified into four groups based on [86]. The **general-purpose verification tools** are "modelling and verifying the protocol using specification languages and verification tools not specifically developed for the analysis of cryptographic protocols" [87]. In the **developing expert systems approach,** unique systems for the actual protocols are created "that a protocol designer can use to investigate different scenarios" [87]**.** The **modal logic approach** is "modelling and verifying the protocol using modal logics developed for the analysis of knowledge and belief" [87]. Furthermore, the **algebraic approach**, which is "developing a formal model based on the algebraic term-rewriting properties of cryptographic systems" [87].

In this dissertation, the **Applied-π calculus** is used for modelling, and the **ProVerif** software is for automatic verification of the designed protocol, presented in Section 3.2.3.2. Applied-π calculus is a Dolev-Yao type model, and ProVerif is an algebraic verifier. A brief introduction to the Applied-π calculus and ProVerif software tool is given in this section.

The Applied-π calculus is a particular quasi-programming language, which was made for modelling cryptographic communication protocols and checking security properties, by Abadi and Fournet and introduced in [88]. Applied-π calculus is the extension of the **π-calculus** [89] and is close to the **spi-calculus** [90]. Adding a rich term algebra to the original π-calculus – which was made to model concurrent processes – gives the advantages to the applied-π calculus to be able to formalize cryptographic primitives. Applied-π calculus offers comprehensive collections of complex cryptographic primitives, like digital signatures, hash functions and proofs of knowledge. The calculus has been used in the analysis of several types of protocols, like certified email [91], electronic voting [92] or authentication protocols [93]. In this model, the terms represent messages of the protocol, function symbols represent cryptographic primitives, and the properties of the primitives are modelled by equations [87]. In Table 14. the necessary notations of the calculus are summarized, for detailed information see [88].

Table. 14. Basic notations in Applied-π calculus

Denotation	Explanation
$L, M, N, T, U, V ::=$	Terms
$a, b, c, \ldots, k, \ldots, m, n, \ldots, s$	Names
x, y, z	Variables
$f(M_1, \ldots, M_l)$	Function application
$P, Q, R ::=$	Processes (or plain processes)
0	Null process
$P \mid Q$	Parallel composition
$!P$	Replication (infinite number of copies of P running parallel)
$\nu n. P$	Name restriction ("new") (makes a new, private name n then behaves as P)
$if\ M = N\ then\ P\ else\ Q$	Conditional
$u(x). P$	Message input
$\overline{u}\langle N \rangle. P$	Message output

ProVerif is a software tool, which is based on the Applied-π calculus, the syntax is really close to it. ProVerif can analyze properties' reachability, correspondence assertions, and observational equivalences automatically. With the help of ProVerif, it is possible to formalize different cryptographic primitives, like encryption and decryption functions, digital signatures or hash functions. If ProVerif results that a property is satisfied, then the model guarantees the property, but ProVerif may not be able to prove a property that holds. For security evaluations, ProVerif uses queries that might be a fact or a correspondence.

In the implementation of the designed „location-stamping" protocol events are defined to mark the examined stages of the process, and then the relation between those events is tested. Like for example, whether an event a has been executed, then event b has been executed previously. In this section, security requirements, authenticity and data integrity are examined. Hereunder in Table 15. the necessary notations of the ProVerif can be seen, for detailed information consider examining [94].

Table. 15. Basic notations in ProVerif

Denotation	Explanation
a, b, c, k, m, n, s	Names (a, b, c usually denotes channels)
x, y, z	Variables
$const\ c.$	Constant (c is the name of the constant)
$fun\ f/n.$	Constructor (f is the name the constructor, n is the number of its arguments)
$reduc\ g(f(m), k) = m.$	Destructor (g is the name of the destructor; m is the argument of the constructor and k is the argument of the destructor)
$free\ a$	Channel
let	Keyword to create processes
new	Keyword to generate new value to variables
$event\ e(x)$	Event (e is the name and x is the argument of the event)
$in(a, m)$	Replication (infinite number of copies of P running parallel)
$query\ ev: e_1(x) \Rightarrow ev: e_2(y)$	Correspondence assertion (for each occurrence of $e_1(x)$, there is a previous occurrence of $e_2(y)$ for some y, where $e_1(x)$, $e_2(y)$ are events)
$query\ evinj: e_1(x) \Rightarrow evinj: e_2(y)$	Injective correspondence (correspondence assertion for one to one relationship)

Below the code of the higher safety „location-stamping" protocol can be seen implemented in ProVerif. First, it is necessary to declare the used channels, cryptographic primitives and other calculator functions of the protocol. As you can see the channel *a* is declared as a free channel, which means it is lack of any security. So, any attacker is able to get or maliciously modify the messages that goes through it. A few functions are declared here, like a hash function *h*, a digital signature key generating function *pk*, a digital signing function *sign*, a digital signature verifier function *checkSign* and *getMess*, and two functions in order to calculate the actual GPS coordinates from the received raw data the *calc* and its verifier the *checkCalc* functions.

Declarations

```
(*Channel*)
free a.
(*Hash function*)
fun h/1.
(*Digital signature*)
fun pk/1.
fun sign/2.
reduc checkSign(sign(m,k),pk(k)) = m.
```

```
private reduc getMess(sign(m,k)) = m.
(*Coordinate calculator function*)
fun calc/1.
reduc checkCalc(calc(RD)) = RD.
```

So, in the implementation of the task done by the two participants – Mobile Device and Certification Authority – *a* is a public communication channel on which the participants can communicate, *h* is the hash function, which is used to shorten the input messages. The *pk* method is defined to generate a public key for a digital signature process, with a secret key as an input parameter. The digital signature process is denoted by *sign*. To be able to check the correctness of the digital signatures, the *checkSign* function is created and to get back the original message from the signed message, the *getMess* function is used. The *calc* function can calculate accurate GPS coordinates from the raw data given by the satellites, and *checkCalc* is able to verify the result of the calculation of the GPS coordinates. One main process and two subprocesses – one for each participant – have to be created. In this proof, a reliable Mobile Device is considered.

Mobile Device Process

```
let processMD =
   new m;
         event startedMD(sskMD);
   let h1 = h(m) in
   let s1 = sign(h1,sskMD) in
   new RD;
   let h2 = h(RD) in
   let s2 = sign(h2,sskMD) in
   let GPSc = calc(RD) in
   let h3 = h(GPSc) in
   let s3 = sign(h3, sskMD) in
   let h4 = h((h1,RD,GPSc,s1,s2,s3)) in
   let s4 = sign(h4,sskMD) in
   out(a,m);
   out(a,(h1,RD,GPSc,s1,s2,s3,s4));
         event requestMD(h1,RD,GPSc,s1,s2,s3,s4);
   in(a,(ALS,t2));
        event getALS(ALS,spkCA);
   if getMess(ALS) = (h((h1,RD,GPSc,s1,s2,s3,s4)),t2) then
         event DataTheSameMD();
      if checkSign(ALS,spkCA) = (h((h1,RD,GPSc,s1,s2,s3,s4)),t2) then
                              event AcceptSignatureMD(spkCA).
```

Mobile Device Process starts with creating a new message and then follows the process of the higher safety level protocol explained earlier in Section 3.2.3.2. In the Mobile Device Process five events are defined, the following table shows the purpose of it.

Table. 16. Events in the Mobile Device Process

Name	***Parameter***	***Explanation***
startedMD/1	*The secret key of the Mobile Device*	*Marks that the Mobile Device gets a new message to create a „location-stamp".*
requestMD/7	*Hash and original values of the signed message, raw data, and GPS data.*	*Marks that the Mobile Device asked for a new „location-stamp" for the Certification Authority.*
getALS/2	*Authenticated „location-stamp" and the public key of the Certification Authority.*	*Marks that the Mobile Device gets the Authenticated „location-stamp" from the Certification Authority.*
DataTheSameMD/0		*Marks that the „location-stamped" message is identical to the original one.*
AcceptSignatureMD/1	*The private key of the Certification Authority.*	*Marks that the Certification Authority made the signature.*

Certification Authority Process

```
let processCA =
        in(a,m);
                event startedCA(spkCA);
        in(a,(h1,RD,GPSc,s1,s2,s3,s4));
                event getRequest(h1,RD,GPSc,s1,s2,s3,s4);
        if getMess(s1) = h1 then
                if getMess(s2) = h(RD) then
                        if getMess(s3) = h(GPSc) then
                                event DataTheSameCA();
if checkSign(s2,spkMD) = h(RD) then
                event AcceptSignatureCA(sskMD);
```

```
                if checkCalc(GPSc) = RD then
                        event AcceptCoordinateCA(RD);
let h5 = h((h1,RD,GPSc,s1,s2,s3,s4)) in
        let ALS = (sign((h5,t2),sskCA)) in
        out(a,(ALS,t2));
                event sendALS(ALS,sskCA).
```

Certification Authority Process goes through the steps of the higher safety level protocol explained in Section 3.2.3.2. Events of the Certification Authority Process explained below.

Table. 17. Events in the Certification Authority Process

Name	***Parameter***	***Explanation***
startedCA/1	*The secret key of the Certification Authority*	*Marks that the Certification Authority gets a new message from the Mobile Device to create a „location-stamp" on it.*
getRequest/7	*Hash and original values of the signed message, raw data, and GPS data.*	*Marks that the Certification Authority gets the request with the right properties.*
DataTheSameCA/0		*Marks that the message sent by the Mobile Device is the same, which is in the request.*
AcceptSignatureCA/1	*The private key of the Mobile Device.*	*Marks that the message sent in the request is identical as the original one.*
AcceptCoordinateCA/1	*Raw satellite data.*	*Marks that the GPS coordinate sent in the request can be calculated from the raw data sent in the request.*

Main Process

```
process
  new sskMD;
  new sskCA;
  let spkMD = pk(sskMD) in
  out(a,spkMD);
  let spkCA = pk(sskCA) in
  out(a,spkCA);
  ((!processCA)| (!processMD))
```

Implementing the two processes for the participants makes it possible to create the main process. The protocol starts to run in the main process; the Mobile Device and the Certification Authority processes are called here. To be able to start running two secret keys have to be generated, *sskMD* is the secret key for the Mobile Device process, and *sskCA* is the secret key for the Certification Authority process. Corresponding public keys – *spkMD* and *spkCA* – then generated with the use of the *pk* function. Both public keys are published on the free channel *a*, allowing the opportunity to both participants to verify the source of the message, by verifying the digital signature made with the secret keys generated in this main process. The Certification Authority and the Mobile Device processes are called as parallel processes and repeat continuously.

Proved properties

Correctness proof with Applied-π, and ProVerif based on the declared events in the processes, here the correspondence between the earlier declared events are investigated. Two types of relations can be examined, the correspondence assertion and the injective correspondence. Correspondence assertion is a 1:N (one to many) relation, while injective correspondence is 1:1 (one to one). The fundamental difference between the two types of correspondence that the injective one can prove only if an event occurred then the other event occurred too, correspondence assertion on another hand is able to prove how many times are one of the events occurred if the other one has occurred. To prove authenticity and data integrity correspondence assertions and injective correspondence are checked for events declared in the subprocesses of the participants.

Authenticity is the property that proves that the origin of some information is the one who claims it to be as it is defined in Section 2.

Definition 1.
Let us state that our protocol fulfils authenticity if the following conditions hold: The

Mobile Device successfully authenticates the Certification Authority and the Certification Authority successfully authenticates the Mobile Device.

Theorem
Higher safety „location-stamping" protocol accomplishes the authenticity property defined above.

Proof
CA and MD authentication are achieved because the following queries return the logical value true:
(*Proof of the authentication of MD to CA*)
query ev:AcceptSignatureCA(x) ==> ev:startedMD(x).
(*Proof of the authentication of CA to MD*)
query ev:AcceptSignatureMD(x) ==> ev:startedCA(x).

Data Integrity is a property that provides that a message is not altered through any process as it is defined in Section 2.

Definition
Let us state that our protocol fulfils Data Integrity if the following conditions hold: The M(message), RD(raw data) and the GPSc(GPS coordinate) sent by the MD has not been tampered through the transmission, and the ALS(authentic „location-stamp") sent by the CA has not been tampered through the transmission.

Theorem
Higher safety „„location-stamping"" protocol accomplishes the Data Integrity property.

Proof
Data Integrity is achieved because the following queries return the logical value true:
(*Proof of data integrity for CA*)
query ev: DataTheSameCA().
(*Proof of data integrity for MD*)
query ev: DataTheSameMD().

Complexity

Determining the computational complexity of a „location-stamping" protocol requires examination of one single run of the protocol according to the number of the performed operations. Six hash prints, seven digital signings, three digital signature verifications, and two GPS coordinates determination from raw data are required to generate a „location-stamp". Furthermore, it is necessary to know the

computational complexity of the used cryptographic primitives, so it has to be considered which hash function and digital signature scheme have the least computational complexity. Based on [95] the computational complexity of the RSA digital signature is $O(n^3)$ and ElGamal digital signature is $O(n^3)$ also, where n is the size of the binary representation of the modulus, which is also the size of the key. The computational complexity of the determination of GPS coordinates from the raw data can be ignored, because the size of the raw input data is always the same, nor the size of the message that has to be „location-stamped" nor the type of the signature or hash function is affected by it. The computational complexity of the different types of hash functions like Message Digest Algorithm (MD) and Secure Hash Algorithm (SHA) is $O(n)$. That means, the complexity of our algorithm is based only on the type of the digital signature. So, higher safety solution „location-stamping" protocol has an $O(n^3)$ computational complexity with wether the RSA digital signature or the ElGamal signature scheme is used.

3.2.3.5 APPLICATION OF THE PROTOCOL

The higher-level protocol is built in a portable electronic device, which has been patented by the University of Debrecen in the US, with the name of "Portable electronic device, system, and method for authenticating a document associated with a geographical location". A summary of the patent application is given here, in detail the technical description can be found in [96]. In a portable electronic device, a method of authenticating a document associated with a geographical location is disclosed. A document is provided in the form of digital data, and a hash value is generated from the digital data of the said document. Raw GPS data are received from at least one GPS satellite, and then digitally signed by a first private key of the portable electronic device. From the raw GPS data, exact GPS coordinates are calculated. A request for an authentic „location-stamp" is sent to a certification unit, the request containing at least the hash value of the document, the raw GPS data, and the exact GPS coordinates, wherein said request is digitally signed by a private key of the portable electronic device. In response to said „location-stamp" request, a nonce value from the certification unit is received, said nonce value is digitally signed by a private key of the certification unit. A certification request is then sent to the certification unit, said request containing at least the hash value of the document, the raw GPS data, the exact GPS coordinates and the nonce value, wherein the certification request is digitally signed with said private key of the portable electronic device. In response to said certification request, a certified „location-stamp" containing said certification request and a piece of time information is received, said „location-stamp" being digitally signed by a private key of the certification unit. The certified „location-stamp" is verified by using the corresponding public key of the certification unit, and if it is determined that the

certification unit signs the certified „location-stamp”, the certified „location-stamp” will be assigned to the document. The basic structure of the system and the operation flow of the algorithm is displayed in Figures 19., 20. and 21.

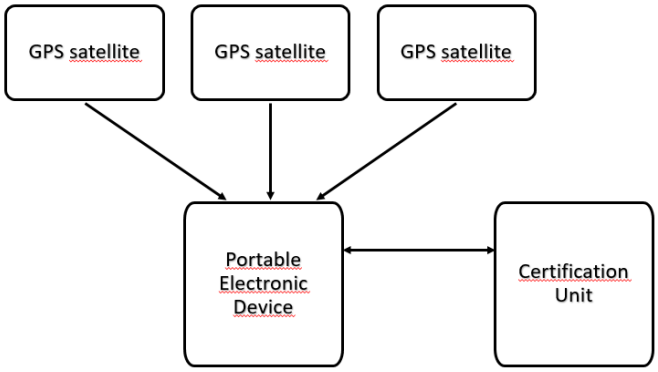

Figure 19. Structure of the system

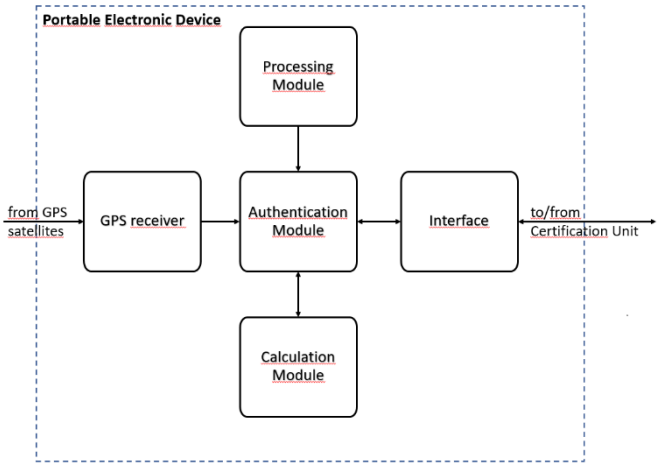

Figure 20. Structure of the Portable Electronic Device

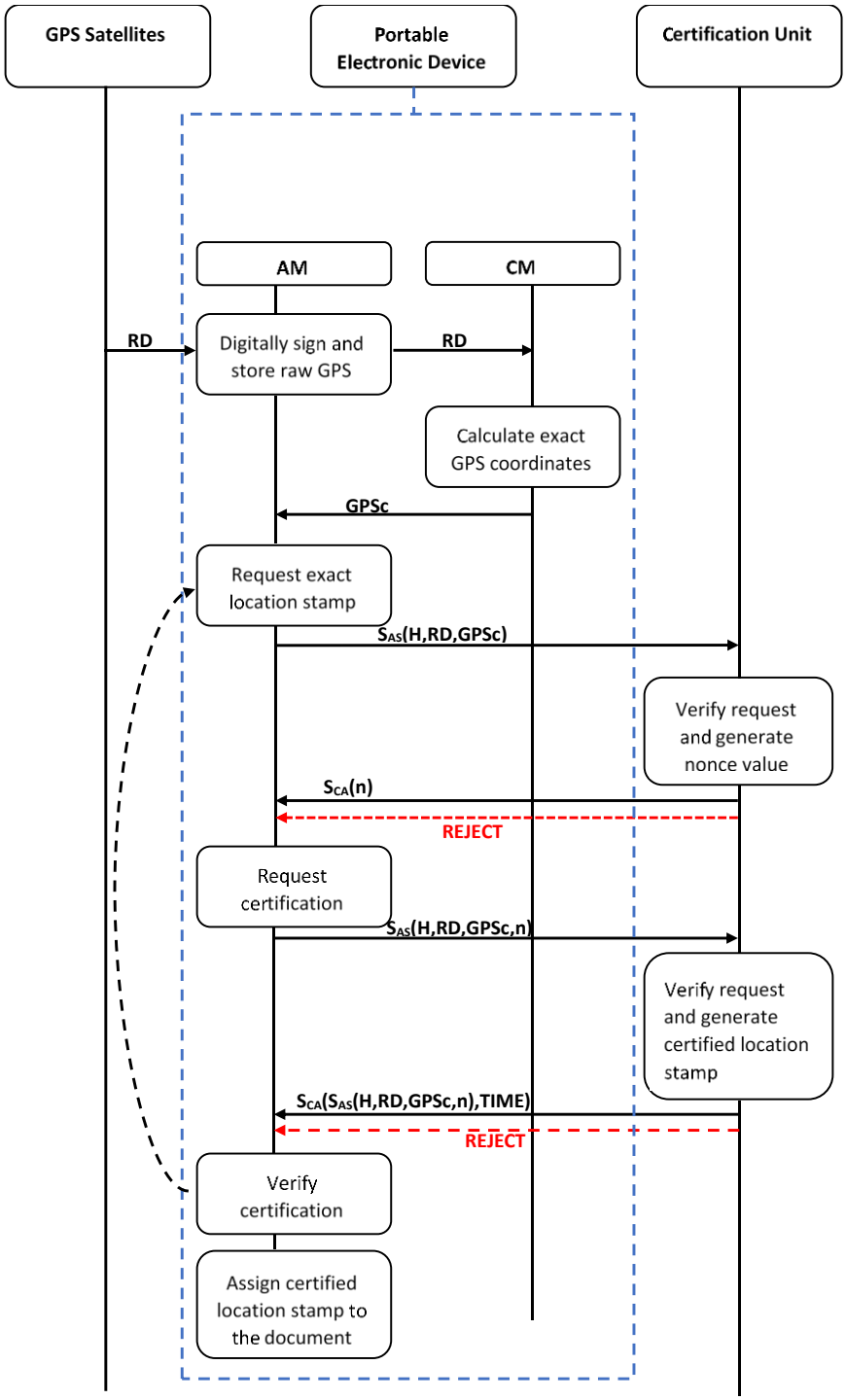

Figure 21. The operation flow of the system

3 MODBUS PROTOCOL

In this section, the security breaches of critical infrastructures – the infrastructures that are crucial for the regular operation of any country – are investigated. Critical infrastructures are supervised with SCADA systems and here one fundamental building block of such SCADA systems, namely the Modbus RTU industrial communication protocol, is overviewed. The section is focusing on the security problems, and lack of security of Modbus protocols and a secure Modbus RTU protocol is presented.

4.1 GENERAL DESCRIPTION AND APPLICATIONS

4.1.1 GENERAL DESCRIPTION

"The **Modbus** serial communications protocol is a de facto standard designed to integrate **PLC**s (**Programmable Logic Controller**), computers, terminals, sensors, and actuators" [97].

Modbus was initially designed and made by Modicon (now Schneider Electric) in 1979 to "establish master-slave/client-server communication between intelligent devices" [98]. Modbus is used in **Industrial Control Systems** (**ICS**), where different processes run in parallel, like reading sensors, controlling actuators, transferring data to operate or automate industrial processes. NIST defines the ICS in [99] like it is "an information system used to control industrial processes such as manufacturing, product handling, production, and distribution. Industrial control systems include **Supervisory Control and Data Acquisition Systems** (**SCADA**) used to control geographically dispersed assets, as well as **Distributed Control Systems** (**DCS**) and smaller control systems using Programmable Logic Controllers (PLC) to control localized processes." ICS can be classified by the functionality, the size and the level of automation. The four main types of ICSs are the PLC, DCS, SCADA and the **Industrial Automation and Control System** (**IACS**). In this dissertation, the focus is on the SCADA systems, more precisely on those SCADA systems which control processes of **Critical Infrastructures** (**CIS**). In 2019 sixteen sectors were marked by the Department of Homeland Security of the USA as critical infrastructures, such as the communication sector, the emergency services sector, the energy sector, the healthcare, and public health sector, the transportation sector, the information technology sector or the water and wastewater systems sector [100].

In 1976 the first distributed loop controller the TDC-2000 controller was introduced by Honeywell, which resulted in the rapid spread of ICSs and started the history of such communication protocols like Modbus. Communication between devices made

by different manufacturers, like sensors, actuators, or terminals in any ICS system requires converters between different physical implementations, agreements on the data encoding, error checking, type of channels and on many other essential properties of the network and indeed a common language for the communication. The ISO (International Organization of Standardization) in 1984 made a model to standardize the communication between different digital devices, namely the **OSI** (Open System Interconnection) **reference model** [101]. The recommendation of the reference model is overviewed in the following introduction. The model contains seven different layers, where every layer is served by the layer underlying, and every layer is responsible for different aspects of the communication process. A common framework is given by the model to the network designers to unify the architecture of the communication. In Figure 22. the layers of the ISO/OSI model can be seen.

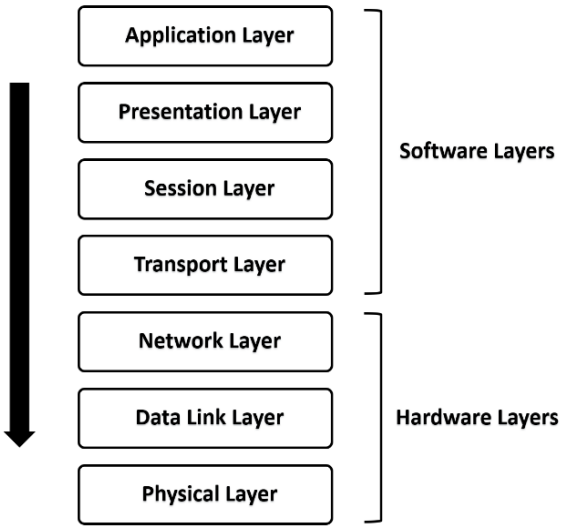

Figure 22. ISO/OSI model

The lowermost layer is responsible for the physical connection, precise parameters of the used physical media; the structure of the network is defined and realized here. It provides mechanical, electrical, functional and procedural methods. The stability of the network is based on this layer because the physical connection is activated, maintained and deactivated here. Two more layers are responsible for physical aspects aka hardware of the network, the data link, and the network layer. The data link layer provides error and access control over the physical layer. Routers and gateways are set up by the network layer, and several configuration parameters are defined here. Transport, session, presentation and application layers deal with the software structure of the network. The transport layer protocol provides optimization of the available services. The session layer synchronizes and manages the data exchange, while the purpose of the presentation layer is to provide a

common representation of the data transferred by the applications. Application is the uppermost layer of the ISO/OSI reference model and deals with the data exchange between applications directly.

A certain number of communication protocols were designed during the past half-decade to solve the communication between different devices in ICSs. At the time of the spreading ICSs, the communication protocols which standardized the communication in the abovementioned network started to appear. In the beginning serial bus connection was used as an underlying physical media, and the early protocols mostly give recommendations only for the physical and data link layer of the ISO/OSI reference model. Later the Ethernet protocol [102] was made as a standard for the two lowermost layers of the ISO/OSI reference model. The recommended underlying physical media has changed, and both parallel and serial connections can be realized with the use of Ethernet. Based on [103] in 2017, **serial bus protocols** serve 48%, and **Ethernet-based protocols** serve 46% of the worldwide industrial network usage. The most important and widely used serial protocols are chronologically the **Modbus RTU** (1979), the **Profibus** (1987), the **DeviceNet** (1994) and the **CANopen** (1995). All the beforementioned protocols adapted to the changes and can be used based on Ethernet standard too; these protocols called **Modbus TCP/IP**, **Profinet, Ethernet/IP** and **EtherCAT COE** respectively. In Figure 23., the relative popularity of serial bus and ethernet protocols can be seen.

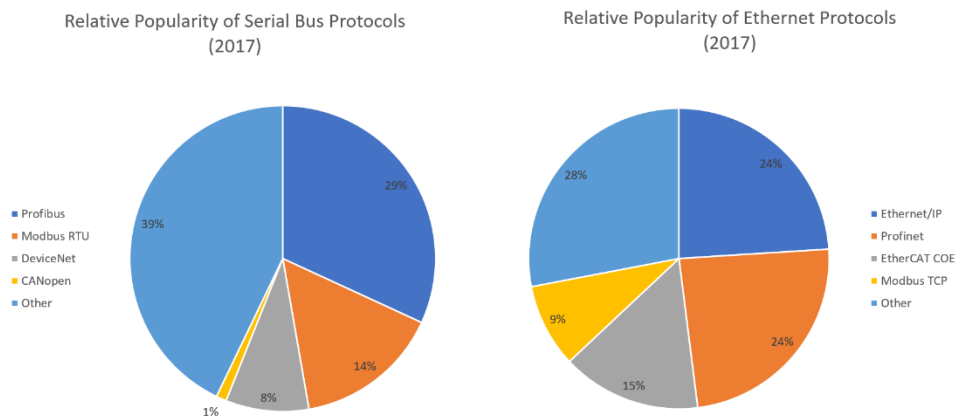

Figure 23. The relative popularity of industrial protocols [103]

The previously mentioned communication protocols realize the layers of the ISO/OSI reference model, but usually not all the seven. Serial Modbus collapsed into three layers, the physical, the data link, and the application layers are defined only. Ethernet itself contains the physical and the data link layer; thus, Ethernet-based Modbus defines the two lowermost layers, the network layer, and the application

layer. Profibus uses the same three layers like serial Modbus, and Profinet realizes the network and the transport layers additionally. DeviceNet implements physical, data link, transport, and network layers, upper layers are provided by the CIP (Common Industrial Protocols), Ethernet/IP implements all the seven layers. CANopen uses the physical, the data link, the transport, and the application layers of the ISO/OSI in the case of EtherCAT COE only the presentation and the session layers are not implemented.

As it is mentioned earlier communication protocols as Modbus serve the primary basis for the ICS system, especially for SCADA designed to supervise and maintain critical infrastructures. In the following, the technical documentation of the Modbus protocol is overviewed [104], [105], [106]. The Modbus standard defines three modes for digital devices in a SCADA to communicate. The three different Modbus standards are the **Modbus RTU**, the **Modbus ASCII**, and the **Modbus TCP/IP**, RTU, and ASCII modes are part of the serial line standard, while TCP/IP mode is ethernet based. Modbus RTU and Modbus ASCII are referred to as Modbus Over Serial Line and Modbus TCP/IP as Modbus Application Protocol in the technical documentation. RTU and ASCII modes are defining the data link layer of the ISO/OSI reference model, while the TCP/IP mode is defining the application layer of it. The connection between Modbus and ISO/OSI reference model can be seen in Figure 24.

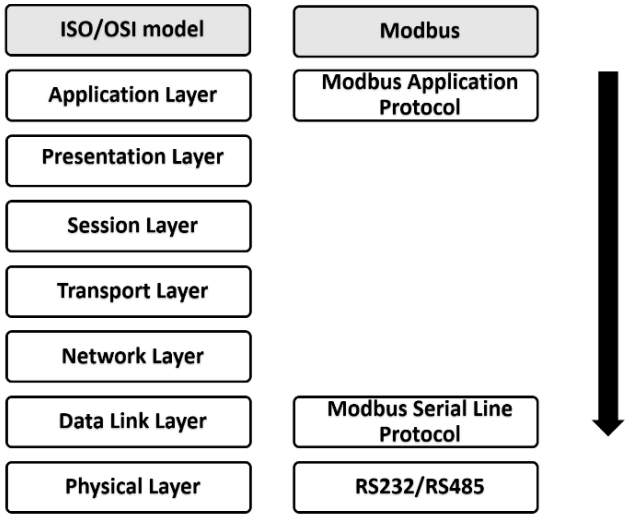

Figure 24. ISO/OSI reference model and Modbus

4.1.1.1 MODBUS OVER SERIAL LINE

The serial line Modbus protocol is located on the data link layer of the ISO/OSI reference model. Under serial communication, a bit by bit transmission of the data is meant, on a single path over a cable. The underlying physical connection is realized

over RS-232 (Recommended Standard 232) [107] or RS-485 (Recommended Standard 485) [108] data transmission standards. Data transmission standards such as the beforementioned, define important properties, like voltage levels (line voltage, received signal voltage, operation voltage), line impedances, baud rate, underlying media, or type of connectors. Shielded or unshielded twisted pair cables can be applied, and there is no specific connector designed to the standard. Simple, half-duplex, and full-duplex connections can be realized. While RS-232 is able to bridge at most 15 meters, the RS-485 up to 1200 meters.

Serial line Modbus is a **Master-Slave protocol**, thus on a serial bus, a single Master node – at a specific time – and up to 247 Slave nodes can be connected. The Master node controls communication. **Request-response communication pattern** is used, under what the following communication process is meant. Every connected Slave node has a unique address on the bus; it is a number from 1 to 247. Only the Master device can initiate a request, two addressing modes can be used, namely unicast and broadcast mode. The two addressing modes displayed in the following figure.

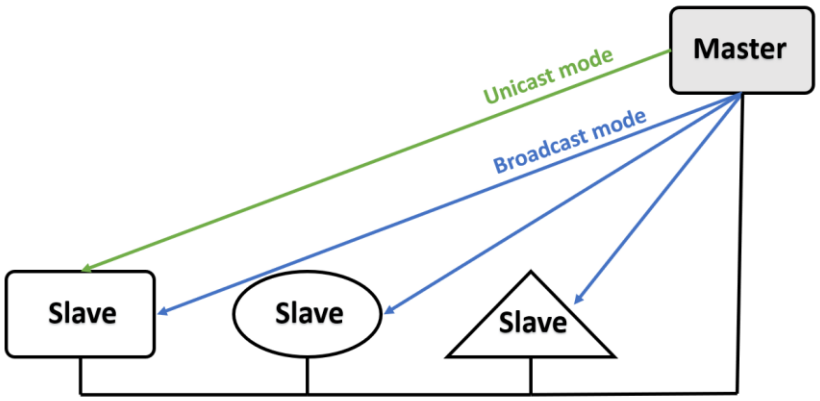

Figure 25. Unicast and broadcast mode in Modbus over Serial Line

In **unicast mode**, one individual Slave node is addressed at a time, and in broadcast mode, the 0 address is used to address all the Slave nodes at the same time. Every Slave node has to be able to recognize the 0 address, which as it is mentioned before, reserved for the **broadcast mode**. Slave nodes cannot initiate requests, only able to respond to the request of the Master node, although the response has to be addressed to the Master. There is no address assigned to the Master node because only one Master can be installed on a network like the above. Slave nodes cannot communicate with each other.

Modbus protocol is using a specific frame structure to standardize the communication over a serial line. The frame contains an address field, a function code, the data and an error checking field as it is displayed in Figure 26.

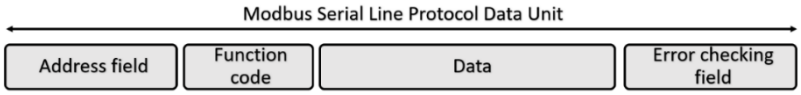

Figure 26. Modbus Serial Line PDU

The **address field** contains the address of the individual Slave node if the Master is in unicast mode, or 0 if the Master is in broadcast mode. **Function code**s determine the actions that have to be performed by the Slave node. Function code 01 is, for example, force the addressed Slave device to read the ON/OFF status of discrete coils or function code 17 returns with a description of the addressed Slave node. Valid function codes are in the 1-255 range. The **Data field** contains parameters that are necessary to process the requests or responses. Parameters like, the amount of the coils that have to be checked for ON/OFF status or the number of the first coil in the case of function code 01. **Error checking field** provides integrity validation method for the Slave nodes and confirmation of message validation for the Master.

The Modbus data link layer differs in the transmission mode in the case of Modbus RTU and Modbus ASCII. The two transmission modes define the content of the Modbus Serial Line Protocol Data Unit (PDU) a.k.a. the Modbus frame differently. Before sending, transmitted Modbus message is forced into a specific frame by the transmitter device. Transmission mode on one serial bus has to be the same for every device at a certain time. In the case of Modbus RTU Slave 1 byte is reserved for both the address and the function code, 2 bytes for the error checking field and at most 252 bytes for the data field of the PDU. In Modbus ASCII 2 characters are reserved either for the address, the function code, and the error checking field and up to 2x256 characters for the data field of the PDU. The frames of the two transmission modes are displayed in Figures 27. and 28.

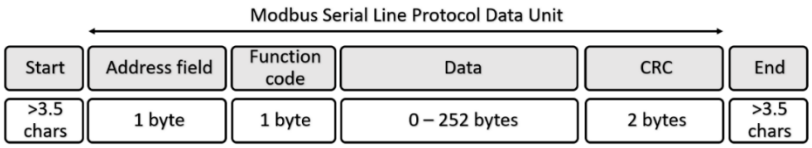

Figure 27. Modbus RTU frame

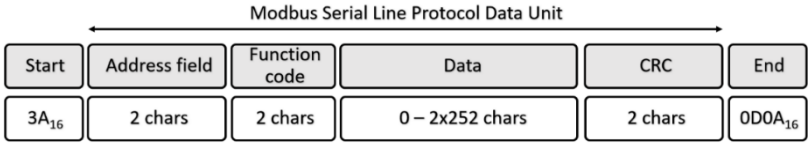

Figure 28. Modbus ASCII frame

Either the starting or ending points of a single frame is denoted by at least 3.5-characters times long silence in the case of Modbus RTU if the silent interval is longer than 1.5-characters times during the transmission the frame is considered incomplete. Modbus ASCII denotes the start of the frame with one character especially the hexadecimal ASCII code of the ‘colon’ and the ending point with two special characters, namely the pair of hexadecimal ASCII values of ‘carriage return’ and ‘line feed’. If the silent interval is longer than 1 second, then it is considered as an error in the actual transmitted frame.

Transmission mode differs in not just the structure and content of the Modbus frame but the method applied for error checking. **Cyclical Redundancy Checking** (CRC) is used in Modbus RTU and **Longitudinal Redundancy Checking** (LRC) for Modbus ASCII. Both CRC and LRC values are calculated and appended to the original message by the transmitter and recalculated and compared by the receiver. Calculated and recalculated values have to be the same. CRC field contains a 16-bit binary value, which is calculated following the steps below:

1. Fill the 16–bit long CRC register with ‘1’-s.
2. XOR the first 8–bit of the message with the first 8-bit of the CRC register, then put the result into the CRC register.
3. Shift the CRC register one bit to the right and add a 0 to the most significant bit.
4. If the just deleted least significant bit was 0, then repeat the third step. Otherwise XOR the CRC register with 1010 0000 0000 0001.
5. Repeat the third and fourth steps until eight shifts have been performed.
6. Repeat steps second, third, fourth and fifth for the next eight bits of the message. Continue doing this until all bytes have been processed.
7. The final content of the CRC register is the CRC value.
8. When the CRC is placed into the message, its upper and lower bytes must be swapped. [104]

LRC contains an 8-bit binary value, which is generated by executing the following steps:

1. Add all successive eight bits in the message into an eight bits field, except for the starting and ending characters, and every carrying bit has to be discarded.
2. Calculate the twos–complement of the final eight bits field. [104]

Data integrity is supposed to be provided by the CRC and LRC between the Master and the Slave. Both CRC and LRC are hash values of the message in the sense of Section 2. However, without a cryptographically secure channel, the beforementioned methods prove only that the content of the message is not altered between the claimed Master and the addressed Slave and vice versa. Neither the Slave nor the Master can satisfyingly authenticate themselves, and a secure channel between the participants cannot be set, because of the lack of security considerations in the Modbus over Serial Line Protocol.

4.1.1.2 MODBUS TCP/IP

Modbus TCP/IP is located on the application layer of the ISO/OSI model, and the technical specification has stated [105] that it provides communication between devices connected over different industrial buses. On the transport layer, the Transmission Control Protocol (TCP) on the network layer, the Internet Protocol (IP) forms a basis for communication. Physical and data link layers of the ISO/OSI reference model can be implemented by the Modbus Serial Line Protocol and by Ethernet, although the latter is the recommended and more commonly used. Modbus protocol is using a specific frame structure to standardize the communication between applications, this frame is independent of the different implementations of the lower layers. However, the PDU part of the frame is inherited from the Modbus over serial line protocol. That PDU contains only a function code and the data as it is displayed in Figure 29.

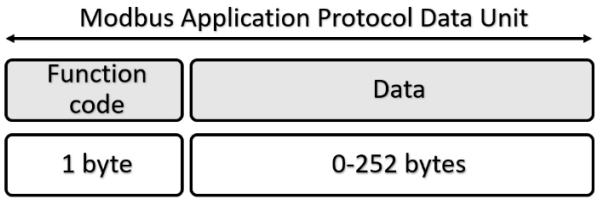

Figure 29. Modbus Application Protocol PDU

Function code is represented on one byte; data field can contain at most 252 bytes. The Modbus TCP/IP frame contains several additional information, the transaction identifier, the protocol identifier, the length, and the unit identifier in the given order. The beforementioned extra fields constitute the Modbus Application Header (MBAP). MBAP and the PDU together form the message frame of the Modbus

TCP/IP, namely the Modbus TCP/IP Application Data Unit (ADU). Modbus TCP/IP is a client/server protocol, where the communication is set up by the client, and the server is only able to respond to those requests until the client closes the connection. Multiple transactions are enabled for the client, but servers can communicate with only one client at a time, concurrent communication is denied for the servers. Modbus ADU is displayed in Figure 30.

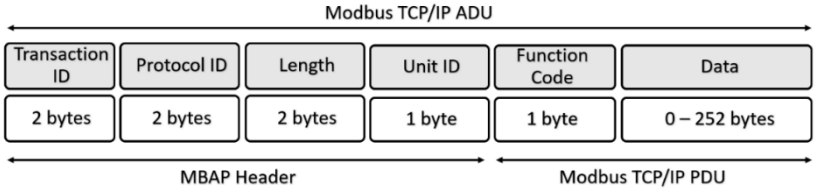

Figure 30. Modbus TCP/IP ADU

In a message, the transaction identifier is used to identify/pair requests and responses. The protocol identifier is always 0 – this field identifies the Modbus protocol –, the length field contains the number of bytes of the remained part of the ADU. The unit identifier identifies slaves on the underlying serial bus network, and function code determines the actions that have to be executed on the server-side of the communication, while the data field contains information that is indispensable to be able to complete the requested actions. For example, the length of the data field or the address of the necessary registers, although the data field can be empty as well.

4.1.1.3 General description of the SCADA system

At the beginning of this section, the **SCADA system** was briefly defined, more detailed definitions of SCADA are quoted below. In [109], Boyer defines SCADA as "the technology that enables a user to collect data from one or more distant facilities and/or send limited control instructions to those facilities." In the IEEE standard [110] SCADA is "a system operating with coded signals over communication channels so as to provide control of RTU (Remote Terminal Unit) equipment." Carke et al. in [111] state that "SCADA refers to the combination of telemetry and data acquisition. SCADA encompasses the collecting of the information via an RTU (Remote Terminal Unit), transferring it back to the central site, carrying out any necessary analysis and control and then displaying that information on a number of operator screens or displays. The required control actions are then conveyed back to the process." As it can be seen from the beforementioned definitions SCADA systems are developed to monitor, collect real-time data, to control system performance, detect and handle abnormalities, manage different devices in industrial systems. Like the water-,

wastewater pipelines, electric power distribution networks, building automation systems, industrial process automation systems, manufacturing, and public transportation.

SCADA system consists of different components, such as the **Operator**, the **Human Machine Interface** (HMI), the **Supervisory System**, the **Communication Network**, the **Remote Terminal Units** (RTU), the **PLC**s, the **Sensor**s, the **Actuator**s and other types of equipment based on [112]. The main task of the Operator is to react on the system feedbacks via HMI, what is that part of the system which is able to report and display the system status and provide functionality for the Operator to intervene. The Supervisory System can be one single PC or several servers depending on the size of the system, and it behaves as a server between the HMI and the RTU and PLC. Under Communication Network, the underlying physical media or wireless connection and the applied communication method are meant. RTUs and PLCs implement the connection between field devices and HMI. Control signals are sent, and the RTU and PLC collect the required data. Sensors and Actuators are field devices, which carry out the measurements and required actions. Sensors are responsible for measuring different characteristics of the system. Actuators, however, are planted into the system to change desired characteristics. In Figure 31. the architecture of a typical SCADA system can be seen.

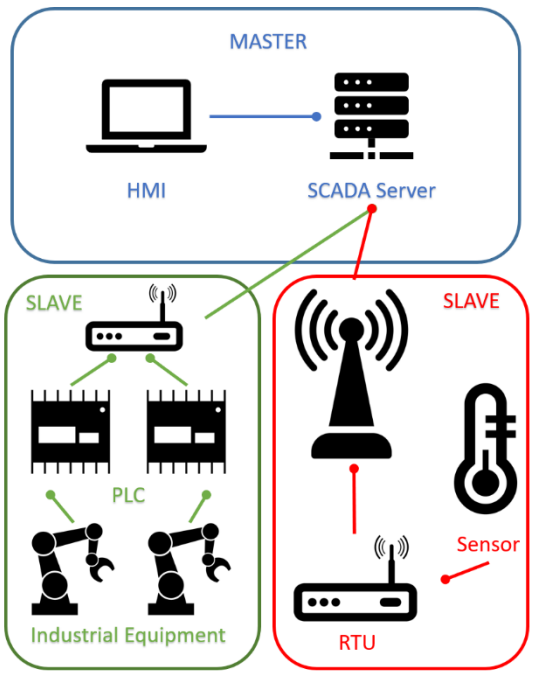

Figure 31. Typical SCADA infrastructure adapted from [113] (icons from Nounproject.com)

4.1.2 APPLICATIONS RELATED TO MY RESEARCH

Modbus is typically used in SCADA systems, which maintain oil, gas, water-, wastewater pipelines, electric power distribution networks, building automation systems, industrial process automation systems, manufacturing, public transportation systems, and healthcare systems. SCADA systems mentioned above vary in size from a low number of participating equipment to large facilities.

In this section, three different sized SCADA systems are presented. The first example is based on an idea published in [114]. This SCADA system is able to supervise the drug infusion pump connected to a patient remotely. In this system, a drug infusion pump and sensors are connected to the human body, where the pump serves as an actuator. Sensors are there to monitor the health status of the patient, measuring different vital signs. Actuators like the drug infusion pump are there to change the condition of the patient for example with the injection of the right amount and right type of drugs, based on the vital signs measured by the sensors. If such a patient's cure is supported by the pump, who is suffering from diabetes, then the pump is responsible for infusion of insulin, and the sensors measure the sugar level of the blood, the pulse and the blood oxygen. The amount of injected insulin can be changed by the pump automatically, by the patient via a PDA or by the doctor via the healthcare system. All options require the analysis of the data collected by the sensor, which is displayed by the Health Monitor PDA or by the HMI. A SCADA system such as the beforementioned one is designed to simplify and facilitate the healing process. The simplified architecture of the SCADA system above is displayed in Figure 32. Communication between the field devices and the controller is wired, and wireless between the controller and the HMI. Modbus RTU and Modbus TCP/IP communication standards are applied in the system. In the simplified structure, only those connections are labelled where the Modbus protocol is used.

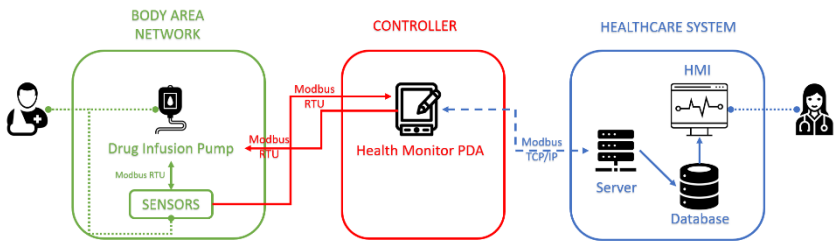

Figure 32. Simplified architecture of a Healthcare SCADA system

The second example is a SCADA system which is supervising the automation of a smart building of the Mechatronics Research Centre of the Faculty of Engineering of the University of Debrecen. The researchers of the Centre have designed and constructed an intelligent and sustainable building. The principal aim of this SCADA

system is to monitor the energy consumption of the building and keep all the properties of it on the desired level to achieve energy efficiency. For the proper functioning of a building like this, many different data are needed. To measure the inner and outer parameters of the research center, sensors are installed, and a comprehensive measurement network is built, which is able to collect and analyze the characteristic of the research center and its environment. Installed actuator devices are able to modify the operating parameters of the building. The SCADA system is consisted of several different sensor and actuator devices, a Sun server to manage the collected data and a PC which serves as the HMI in the system. Sensors are the following: gas-flow meter, heat-flow meter, thermal zone occupancy meter, meteorological station, global solar radiator meter, and an internal wireless sensor network. With the listed sensors the amount of the used gas, the amount of the transferred heat, the number of people at a given time in a given thermal zone, the speed of the wind, the direction of the wind, the duration of the rainfall, the intensity of the rainfall, the external air pressure, the radiation of the sun, the inner temperature, the inner air pressure, and the humidity can be measured respectively. Actuator devices are the rotatable solar collectors, the rotatable solar-cells, the mini wind turbine, and the heat pump system [115]. Based on the data provided by the sensors, the properties of the actuators – like rotation angles of the solar collectors and solar cells, or the amount of electricity generated by the wind turbines – can be changed to keep the building in an energy-efficient state. The simplified structure of the system is displayed in Figure 33. In the simplified structure, only those connections are labelled where the Modbus protocol is used.

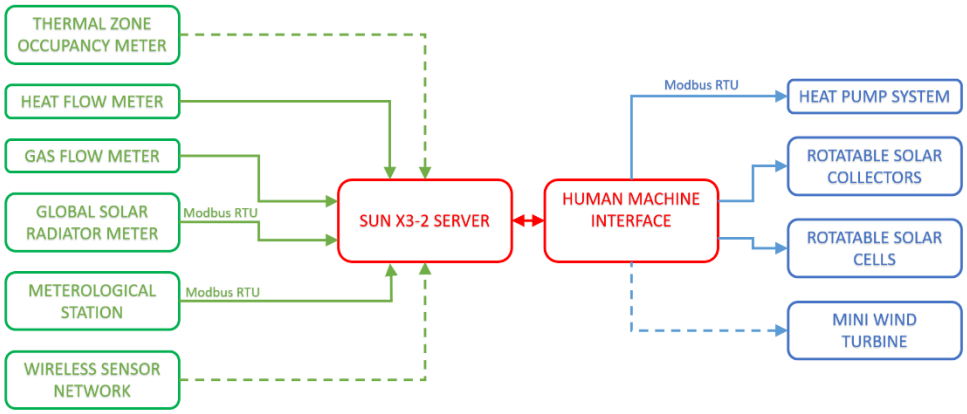

Figure 33. Simplified architecture of the SCADA system of the Building Mechatronics Research Center

The third example is the SCADA system of a water and wastewater plant, which supervises and controls the collection of wastewater and distribution of water [116].

“Water distribution and wastewater collection systems often spread out over large geographical areas. In such systems, elevated water storage tanks and gravity provide most of the energy to create a flow to your faucet through underground pipes. Pumps are used to fill the tanks, and may also be needed at strategic points to ensure adequate pressure in the water distribution system” [117]. SCADA system of this plant contains several HMIs – Central Base Station – which controls or monitors the pumps via RTUs or directly. HMIs can connect to the pump via a serial connection, telephone network, wifi or radio. The Modbus RTU communication protocol realizes serial connection. The presented SCADA system contains several layers because of security considerations, firewalls and backup databases are installed in the system. In Figure 34. that part of the SCADA system which controls the pumps can be seen.

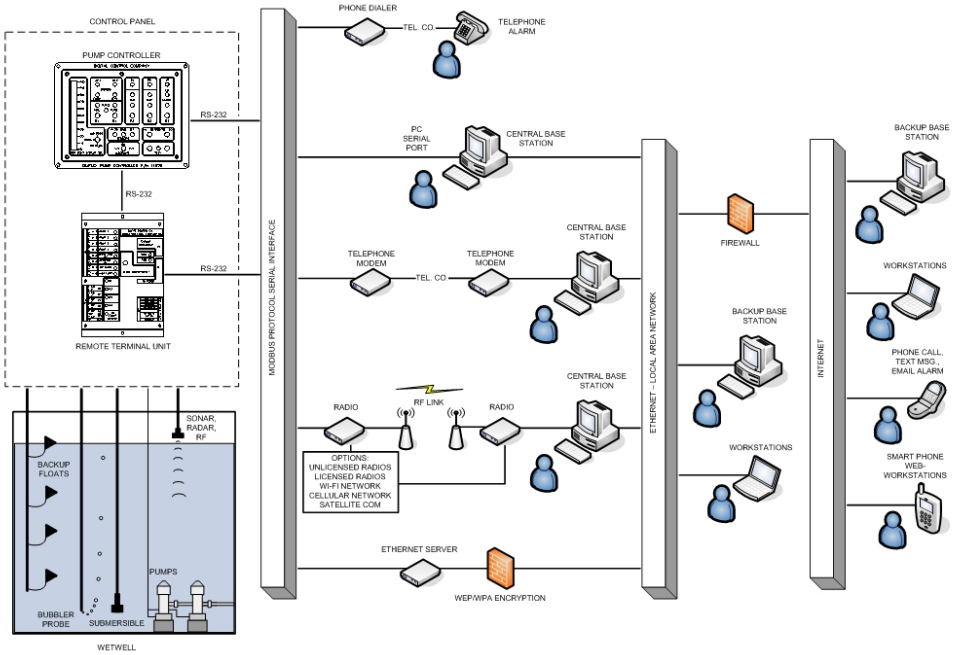

Figure 34. SCADA system of a water and wastewater collector and distributor plant [116]

4.1.3 ATTACKS AGAINST SCADA AND MODBUS

Over time SCADA systems among other ICS systems are increasingly exposed to **cyberattack**s, which can maliciously modify the proper operation of the system or destroy the entire system. Mostly those SCADA systems were targeted by cyber-security incidents, which supervise and maintain critical infrastructures. Consequences of such cyberattacks can not only be financial loss or devastating reputation, but it can be extremely detrimental in health and safety. In this section,

a brief historical introduction is given about the reported cyber attacks that have happened during the last 50 years based on the corresponding scientific literature [118], [119], [120], [121], [122].

There is no consensus on the statement given by Reed in [123] that the explosion which destroyed some part of the Trans-Siberian gas pipeline near Tobolsk in **1982** was a cyberattack or not but in this dissertation, it is considered as the first cyberattack against SCADA systems that was ever reported. The attack based on the injection of a Trojan virus into the SCADA system that controlled the gas pipeline and resulted in an explosion which was equivalent to three kilotons of TNT. According to Reed, the attack was planned and executed by the CIA during the Cold War. The attack was realized through inserting maliciously modified building blocks into the SCADA system, thus locally causing the damage. The financial loss was significant.

In **1992** the emergency alert system of the Chevron Refinery in Richmond California was disabled for 10 hours by a former employee hacking into the computers of the firm [124]. Thousands of people across the USA and Canada were at risk during the outage of the system, because the attack was discovered only when a toxic substance was leaked, and the alert was not generated. The attack was realized through remote access to the system. The ultimate damage magnitude was slight, but lives were threatened.

SCADA system of the Salt River Project – which maintained and supervised about a 210 km long canal system – was hacked via a dial-up modem in **1994** in Phoenix USA. The hacker gained information about the personnel and financial records of the customers and login passwords of employees among many other sensitive data. The attack was realized through remote access to the system. Physical damage magnitude was negligible, but a significant amount of sensitive data were stolen.

With the help of an employee, a Trojan virus is infected the SCADA system of the Gazprom in Russia and allowed hackers to gain access to the control board of the system, which controlled the gas flow in **1999**. The system restored quickly, caused damage magnitude was negligible. The attack was realized through local and remote access to the system.

In **2000** in Australia, a former employee of the Maroochy Water System hacked into the SCADA system, which supervised and maintained the wastewater plant. The hacker afterwards released 1 million litres of sewage into a river and a hotel. The attack was realized through remote access to the system. Significant environmental damage occurred, and the financial and reputational loss was severe too.

In January of **2003**, the Davis-Besse nuclear power plant of Ohio was attacked by a worm named “Slammer”. The SCADA system of the plant was disabled. Fortunately,

the plant was shut down and was in a “safety defueled condition” [125] because of repair. A computer of an employee got infected by a worm via a dial-up modem. The attack was realized through remote access to the system, and no significant damage occurred.

In August of **2003**, an extensive power outage happened in the northeast part of the USA and Ontario. Sixty million inhabitants left without electricity and several critical infrastructures cannot operate in the usual and necessary way. According to Eugen Kaspersky [126] the head of Kaspersky Lab, hackers used malware to take over the control of the SCADA system of the power grid and disabled the warning system. The disabled warning system and some too tall trees caused the complete shutdown of the power grid. The power blackout resulted in several otherwise avoidable deaths and billions of dollars losses. It is not known how the hackers gain access to the system, remotely or locally.

In Poland in **2008**, a juvenile hacked the SCADA system of the tram network, with a DIY infrared remote control. Due to the attack, four trains were derailed, and 12 people got injured.

In **2010** a cyber-attack was planned and executed by the intelligence agencies of the US and Israel. The aim of the attack was not only to gain access over a SCADA system but to delay or halt the Iranian nuclear weaponization, via ruin the uranium centrifuges of the Iranian nuclear plant in Natanz. The worm was planted via a USB stick locally. Stuxnet – the worm used in the attack – infected the computers of the SCADA system in the nuclear plant, and through the computers, those special Siemens PLC-s were connected to the uranium centrifuges. The worm destroyed the centrifuges through increasing the operating time and continuously switching between low and high frequency of the rotors. As a result, the Iranian atomic program was delayed [127], [128].

In **2011** a malware called Night Dragon targeted the SCADA systems of numerous energy plants in Western Europe and the USA. The pishing activity was detected during the attacks, which was originated from China. The attacks were active Chinese time between 9 am and 5 pm for several months [129]. The attacks were realized via remote access, and a significant amount of specific data and intellectual property were stolen.

The number of cyber-attacks continues to grow, and the above presented cases forms only a small portion of the cyberwar. Below the top twelve cyberattacks are listed from **2019** based on [130].

Table. 18. Cyberattacks in 2019

January	"US Department of Justice neutralizes a North Korean botnet used to target critical infrastructure sector and other essential services.
March	Thousands of employees at over 200 oil-and-gas and heavy machinery companies worldwide targeted by Iranian hackers exposing confidential corporate data.
March	Saudi Arabian and US Government industry digital infrastructure targeted by Iranian cyberespionage group.
April	Bayer Pharmaceutical company announced an unsuccessful attack by Chinese hackers.
June	NERC warns that a suspected Russian hacking group was snooping on the electrical utilities' network.
July	Chinese state-sponsored hackers conduct cyberattacks against employees of three major US utility companies.
September	Seventeen US utility companies targeted by a Chinese state-sponsored hacking group.
October	North Korean malware found in Indian nuclear power plant networks.
November	Major manufacturers and operators of ICSs employee accounts targeted by Iranian hackers" [130].

4.2 OWN RESEARCH

4.2.1 REVEALED WEAKNESSES

The Modbus RTU protocol does not include any security considerations. During the design process, the main features of the Modbus protocol were reliability, speed, and accessibility. However, the security of the used channel, thus authentication of the participants, confidentiality, data integrity, and freshness of the messages were left out, so not provided. Our research addressed to find the vulnerabilities of a Modbus RTU based SCADA system. The following table contains the liabilities of Modbus RTU, which are revealed from the related scientific literature.

Table. 19. Vulnerabilities of the Modbus RTU protocol

Lack of Confidentiality [131]	
Lack of Integrity [131]	
Lack of Authentication [131]	
Sensibility for the Man in the middle (MITM) attack [132]	
Sensibility for the Denial of Service Attack (DoS) [133]	
Possibility of Interception [134]	Slave Reconnaissance Modbus Network Scanning etc
Possibility of Interruption [134]	Remote Restart Baseline Response Replay etc.
Possibility of Modification [134]	Diagnostic Register Attack
Possibility of Fabrication [134]	Direct Slave Control etc.

The results of the analysis of the Modbus RTU protocol was summarized with the manageable and straightforward ”Attack Tree Method” [135]. “**Attack tree**s provide a formal, methodical way of describing the security of systems, based on varying attacks. You represent attacks against a system in a tree structure, with the goal as the root node and different ways of achieving that goal as leaf nodes” [135]. The attack tree consists of AND and OR nodes. “Each node becomes a subgoal, and children of that node are ways to achieve that subgoal. OR nodes are used to represent alternatives, and AND nodes are used to represent different steps toward achieving the same goal” [136]. AND nodes are in a parent-children relationship with each other, OR nodes are siblings to each other. To every node, a specific value is assigned, it is ‘P’ (possible) if the attack is possible to realize at the given time, ‘I’ (impossible) otherwise. The value of an OR node is equal to ‘P’ if any of its children assigned value is ‘P’, and ‘I’ if all of its children have the value ‘I’. The value of an AND node is ’P’ only if all children have the value ‘P’, and ‘I’ otherwise.

During the construction of the attack-tree, three possible layers are defined to determine the level of the possible protection. Any Modbus RTU based SCADA system can be protected physically and algorithmically. Under the term physically, acts like locked doors, thick walls, security guards, or hidden slaves and masters are meant.

Physical protection can be divided into two separate parts, external physical protection, and internal physical protection. External protection provides a protected and safe environment for the system. While internal physical protection prevents the physical damage, exchange, or modification of the devices that are parts of the system.

Algorithmic protection means the protection of the applied communication and measurements in the system. The focus is on the algorithmic security of the Modbus RTU protocol, although the attack-tree displays the physical security levels of the Modbus RTU based SCADA system but the methods that can realize such events are not examined in this dissertation., such as the well-known social engineering. The completed attack-tree is presented in Figure 35.

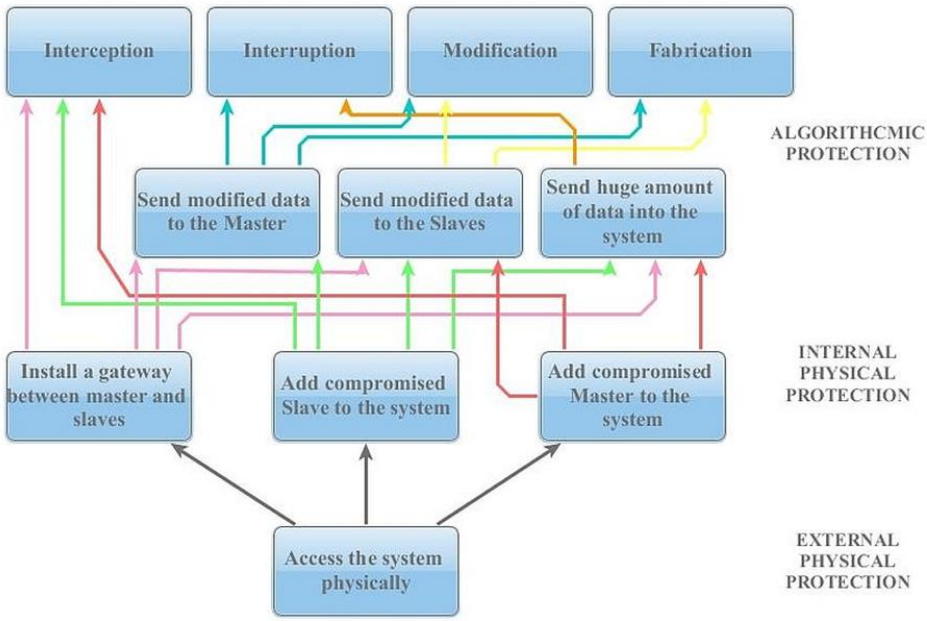

Figure 35. Attack Tree of a Modbus RTU based SCADA system

With this technique, the system can be examined from several points of view. The first viewpoint is to determine the possible purpose of the malicious attacker. The goals of the attacker are narrowed down to four types of attacks; these are the root elements of the attack tree. First is the interception of data like monitoring the channel, then the interruption of the communication - it can be caused by for example a DoS attack -, the third one is the modification of the messages, and the last one is the fabrication of the data. The second viewpoint is to uncover vulnerabilities in the SCADA system, so answering the question, how a malicious attacker gains access to any part of it. The branches of the attack-tree symbolize the revealed vulnerabilities. Based on the completed attack-tree, the attacks can be realized through impacted malicious devices, compromised slaves or masters, all of these attacks are different realizations of the Man in the Middle attack. The paths from the root to the leaves define all the considered attacks against Modbus RTU based SCADA systems, like planting gateways or compromised slave or master devices into the SCADA system. The first root element of the tree is the **interception** of the data. During interception, the attacker captures messages sent over the network, thus acquiring information about the parameters and operation of the system, so the confidentiality of the data and the system is harmed by this attack. The interception can be realized in the following ways, the cases given below are describing three paths from the attack-tree.

- Access to the system physically → Install a gateway between master and slaves → Interception
- Access to the system physically → Add compromised Slave to the system → Interception
- Access to the system physically → Add compromised Master to the system → Interception

The second root is the **interruption**, the goal of the attacker there is to decrease the effectiveness of the system or to manipulate the properties of it, or overload the system to shut it down. Compromising the regular operation of the system is endangering the data integrity and the authenticity of the participants. The interruption can be realized in the following ways, the cases given below are describing three paths from the attack-tree.

- Access to the system physically → Install a gateway between master and slaves → Send modified data to the Master → Interruption
- Access to the system physically → Install a gateway between master and slaves → Send huge amount of data into the system → Interruption
- Access to the system physically → Add compromised Master to the system → Send huge amount of data into the system → Interruption
- Access to the system physically → Add compromised Slave to the system → Send huge amount of data into the system → Interruption
- Access to the system physically → Add compromised Slave to the system → Send modified data to the Master → Interruption

The third root is the **modification**, under what the modification of messages – sent over the channel – by any unauthorized person is meant, harming the data integrity and confidentiality properties. The modification can be realized in the following ways, the cases given below are describing three paths from the attack-tree.

- Access to the system physically → Install a gateway between master and slaves → Send modified data to the Slave → Modification
- Access to the system physically → Add compromised Slave to the system → Send modified data to the Slave → Modification
- Access to the system physically → Add compromised Master to the system → Send modified data to the Slave → Modification

The fourth root is **fabrication**. During the fabrication, a malicious attacker impersonates an authorized user and send fabricated message to the participants of the system. Many properties of security are lost as a result of this attack, like data integrity, the authenticity of the participant and confidentiality. The fabrication can

be realized in the following ways, the cases given below are describing three paths from the attack-tree.

- Access to the system physically → Install a gateway between master and slaves → Send modified data to the Slave → Fabrication
- Access to the system physically → Install a gateway between master and slaves → Send modified data to the Master → Fabrication
- Access to the system physically → Add compromised Slave to the system → Send modified data to the Slave → Modification
- Access to the system physically → Add compromised Slave to the system → Send modified data to the Master → Modification
- Access to the system physically → Add compromised Master to the system → Send modified data to the Slave → Modification
- Access to the system physically → Add compromised Master to the system → Send modified data to the Master → Modification

4.2.2 Existing solutions for the revealed weaknesses

The Modbus protocol is widely used in SCADA systems as it is mentioned in Section 4.1.2. The protocol is comprehensively studied and tested according to the related scientific literature, but the tests are mostly focusing on the Modbus TCP/IP protocol. Modbus RTU is usually examined based on different aims to achieve, like to improve the transmission quality or to detect and correct errors or to find the maliciously modified messages during transmission.

For instance, Urrea et al. in [137], designed a particular modem – that has to be applied in the master and the slave device also – which concatenate **two parity bits** to the messages defined by a **Reed-Solomon code** to be able to detect and correct errors in the messages. “These modems will have the task of receiving the Modbus messages of the associated device, code it, and send the resultant parity bits in the time-spaces through the data bus, as well as receive the Modbus messages of the network recognizing the parity bits, decode the message, and deliver it to its associated device in Modbus format” [137]. Solutions like this are not capable of securing the Modbus RTU protocol cryptographically, the data integrity can be provided, but confidentiality and the freshness of the messages are not. Moreover, neither the sender nor the receiver is authenticated.

In another approach, **intrusion-detection technics** are applied for filtering and finding the maliciously modified messages in a Modbus RTU flow. Intrusion-detection systems raise alerts about suspicious activities for a human operator, where the operator can tune the system by reacting to the notifications. Misuse and anomaly-based intrusion-detection systems can be distinguished, by the first type

known-attacks can be detected, while the anomaly-based systems are able to recognize abnormal behaviour in a system, without any preliminary information about the attacks. Solution published in [138] presents an anomaly-based system, which is “using semantic information about the protocol, such as the type and meaning of a field and expected or observed interdependencies between fields, to construct features that capture the meaning of messages rather than only their format” [138]. Solutions like this are not able to provide any of the required security properties, although the number of malicious actions can be reduced by monitoring the communication flow.

The solutions what is widely applied to secure the communication over Modbus RTU are called “**bump in the wire devices**”. These devices are installed on each side of the communication next to the master and the slaves and serve to enhance the security properties of the communication protocols, such as the data integrity and confidentiality of the messages. In [139] three of these devices are presented, which “predicts the incoming plaintext based on previous observations; compresses, encrypts and authenticates data online, and pre-sends a portion of the ciphertext before receiving the entire plaintext.” Recommended devices are the Yasir [139], the SEL-3021-2 [140] and the ADA SCM [141].

4.2.3 Own solution: realizing cryptographically secure Modbus RTU communication

In the previous section, it is shown that the main problems of communication via the Modbus RTU protocol are the lack of security parameters as confidentiality, data integrity, authentication of master and slave devices, and freshness. The latter causes that slave devices accept a request and perform tasks at any time if the destination address is equal to its address in the Modbus PDU. Furthermore, the master does not verify the origin of the replies it receives. Thus, the problems to be solved are to prevent the messages from eavesdropping, to authenticate the slaves and the master of the network, and to provide data integrity and freshness of the messages sent via Modbus RTU. In the designed solution, it is taken advantage of the fact that the Modbus RTU based communication not always uses the full of its standardized message length and the messages have a relatively low update frequency. Usually, the MTUs and field devices have a built-in AES engine to encrypt the standard request and response before sending it, this engine is built-in by the device manufacturers and has no connection with the applied industrial communication protocol. However, the AES engine alone, with encrypting the messages before sent through the channel is not enough to solve all the revealed problems of the Modbus RTU protocol. Our solution provides authentication of the participants to each other, and data integrity, freshness, and confidentiality of the

messages. In this section, the paper [142] written by my co-authors and me is overviewed.

4.2.3.1 INITIAL CONSIDERATIONS

It is recommended that the initial steps of the secure protocol designed by my co-authors and me be taken when the network is built or for latency systems at the time of regular revisions. The first message exchange has to be taken on a secure channel. On both sides of the master and the slave, random and prime number generators are required to be able to provide proper properties for the secure protocol. The generated prime and random numbers have to meet with the requirements for the minimum lengths recommended by the related scientific literature. All calculations are taken over a finite prime field. A random polynomial over a finite prime field $GF(p)$ is constructed in the initialization, where the following restrictions are applied:

- Order of the field should at least be $NumSlaves^2 + NumSlaves + 1$.
- Generator prime of the finite field p has to be large enough.
- The coefficients of the random polynomials are chosen uniformly from the interval $(1, p-1)$.
- The y values of the random polynomials are computed $mod\ p$ for distinct x values are chosen from $(1, p-1)$.
- The size of y values of the random polynomials has to be equal size to p.

Denotations are listed below.

Table. 20. Denotations

Denotation	Explanation
$\boldsymbol{M}$	Master.
$\boldsymbol{S_i}$	Ith Slave.
$\boldsymbol{NumSlaves}$	Number of the slaves in the network.
$\boldsymbol{P_{ini}(x)}$	A random polynomial over $GF(p)$ used in the initialization.
$(\boldsymbol{x_i^{ini}}, \boldsymbol{y_i^{ini}})$	Ith point of $P_{ini}(x)$ polynomial.
$\boldsymbol{y_i^{ini}}$	$= P_{ini}(x_i^{ini})$
$\boldsymbol{a_i^{ini}}$	Ith coefficient of $P_{ini}(x)$ polynomial.
$\boldsymbol{n}$	Degree of $P_{ini}(x)$
$\boldsymbol{p}$	A large prime number.
$\boldsymbol{prime}()$	Prime generator function.
$\boldsymbol{skey}()$	Secret key generator function.
$\boldsymbol{encrypt}()$	Function for encryption.
$\boldsymbol{decrypt}()$	Function for decryption.
$\boldsymbol{LagrangeIP}()$	Function for constructing secret by Lagrange interpolation.
$\boldsymbol{K_S}$	The secret key of the master.

$\boldsymbol{C_i}$	Challenge value of the ith slave.
$\oplus$	The operator of the XOR
$\boldsymbol{P_{ij}}$	Share.
$\boldsymbol{\|\|}$	Operator of concatenation
$\boldsymbol{Address_i}$	Address of the ith slave.
$\xrightarrow{\boldsymbol{secure}}$	Secure channel.
$\xrightarrow{\boldsymbol{public}}$	Public channel.
$\boldsymbol{SC}$	"Send challenge" function code.
$\boldsymbol{SI}$	"Shares inside" function code.
$\boldsymbol{FC}$	Function code.
$\boldsymbol{TS}$	Timestamp.
$\boldsymbol{EDataRequest}$	The encrypted data part of a request.
$\boldsymbol{DataReq}$	The data part of a request.
$\boldsymbol{EDataResponse}$	The encrypted data part of a response.
$\boldsymbol{DataRes}$	The data part of a response.

Secret key and challenge values – generated in the initialization part – authenticate the slaves to the master, and vice versa and these keys then planted in the devices, so it is only accessible to those specific devices. It is assumed that the AES engine is present in both the slave and the master devices; thus, we recommend at least a 256-bit long secret key. In the test case, which is described in Section 4.2.3.3, a 128 bit-sized secret key is generated and used by each device. Steps of the initialization are the next:

1: M and S_i synchronize the time

2: M generates a large prime number: $\boldsymbol{p = prime()}$

3: M generates a random private key: $\boldsymbol{K_S = skey()}$

4: M constructs a random polynomial over $GF(p)$:

$\boldsymbol{P_{ini}(x) = a_0^{ini} + a_1^{ini}\, x^{ini} + a_2^{ini} x^{ini^2} \ldots + a_n^{ini} x^{ini^n}}$

where: $\boldsymbol{n > NumSlaves}$ and

$\boldsymbol{P_{ini}(0) = a_0^{ini} = K_S}$

5: M selects $NumSlaves$ amount of points (x_i, y_i) on the polynomial, where:

$\boldsymbol{x_i^{ini} \neq x_j^{ini}}$ for $\boldsymbol{i, j = 1 \ldots NumSlaves}$

6: M sends a unicast request consisting the shares to the slaves, and waits for challenges: $\boldsymbol{M \xrightarrow{secure} S_i : (x_i^{ini}, y_i^{ini})}$

request: $\boldsymbol{[i, SC, (x_i^{ini}, y_i^{ini}), CRC]}$

7: S_i receives the share: $\boldsymbol{S_i \xleftarrow{secure} M : (x_i^{ini}, y_i^{ini})}$

8: S_i generates a challenge: $\boldsymbol{C_i = prime()}$

9: S_i sends the challenge to the master: $\boldsymbol{S_i \xrightarrow{public} M : C_i \oplus y_i^{ini}}$

response: $\boldsymbol{[i, SC, C_i \oplus y_i^{ini}, CRC]}$

10: M receives the challenge: $\boldsymbol{M \xleftarrow{public} S_i : C_i \oplus y_i^{ini}}$

11: M calculates C_i : $\boldsymbol{C_i = y_i^{ini} \oplus (C_i \oplus y_i^{ini})}$

12: M sends unicast requests consisting of extra shares to the slaves:

$\boldsymbol{M \xrightarrow{public} S_i : (P_{i1}||P_{i2}|| \dots ||P_{iNumSlaves}) \oplus C_i}$

request: $[\boldsymbol{i, SI, (P_{i1}||P_{i2}|| \dots ||P_{iNumSlaves}), CRC}]$

$\boldsymbol{P_{ik} \neq P_{il} \; for \; any \; k \neq l}$

$\boldsymbol{k, l = 1 \dots NumSlaves^2}$

13: S_i receives the extra shares:

$\boldsymbol{S_i \xleftarrow{public} M : (P_{i1}||P_{i2}|| \dots ||P_{iNumSlaves}) \oplus C_i}$

14: S_i reconstructs K_S from the n pieces of imprints it already has:

$\boldsymbol{K_S = LagrangeIP((x_i^{ini}, y_i^{ini}), P_{i1}, P_{i2}, \dots P_{iNumSlaves})}$

15: M creates an address table about the slaves: $\boldsymbol{(Address_i, (x_i^{ini}, y_i^{ini}), C_i)}$

4.2.3.2 COMMUNICATION

Communication methods differ in the case of different field devices – like sensors and actuators –, however, it is right for both cases, that only the data field of the Modbus RTU frame is encrypted to avoid known plain text attacks. Denotation can be seen in Table 21.

Table. 21. Denotations

Denotation	Explanation
$\boldsymbol{MeasuredData}$	Measured data by the slave.
$\boldsymbol{numberOfShares}$	The number of secret shares.
$\boldsymbol{numShareReg_i}$	Position of the ith register that contains share inside.
$\boldsymbol{numRandomReg_i}$	Position of the ith register that contains random value inside.
$\boldsymbol{numOfUUReg}$	The number of unused, empty registers.
$\boldsymbol{P_{com}(x)}$	A random polynomial over $GF(p)$ used in the communication.
$\boldsymbol{x_i^{com}}$	The ith x value of $P_{com}(x)$polynomial.
$\boldsymbol{a_i^{com}}$	The ith coefficient of $P_{com}(x)$polynomial.
$\boldsymbol{contentReg_i}$	Content of the ith register.
$\boldsymbol{random}(\;)$	Random number generator which returns a single uniformly distributed random number in the interval (0,p-1)

4.2.3.2.1 Communication with actuator slaves

The synchronized time information and a shared secret key are generated in the initialization step. Each slave is aware of the secret key of the master, and the master knows the value of the unique challenge for every slave. The shared information makes it possible that the participants are able to authenticate each other. Confidentiality is provided by the encryption made by the built-in AES engine. Communication description can be seen below:

1: M initiates a request and encrypts the TS and data part of the request with K_S: $\boldsymbol{EDataRequest = encrypt(K_S, DataReq||TS)}$

2: M sends the request to the slave: $\boldsymbol{M \xrightarrow{public} S_i : EDataRequest}$
request: $[\boldsymbol{i, FC, EDataRequest, CRC}]$

3: S_i receives the request: $\boldsymbol{S_i \xleftarrow{public} M : EDataRequest}$

4: S_i makes the checks:

$\boldsymbol{if}\ i = Address_i\ \boldsymbol{then}$ S_i decrypts the data part:

$\boldsymbol{DataReq = decrypt(K_S, EDataRequest||TS)}$

$\boldsymbol{if}$ TS fresh enough $\boldsymbol{then}$

$\boldsymbol{if}$ $DataReq$ is correct $\boldsymbol{then}$ S_i performs the task

$\boldsymbol{else}$ S_i goes back to initial mode

$\boldsymbol{end\ if}$

$\boldsymbol{else}$ S_i goes back to initial mode

$\boldsymbol{end\ if}$

$\boldsymbol{else}$ S_i goes back to initial mode

$\boldsymbol{end\ if}$

5: S_i encrypts the data part of the response and the $(\boldsymbol{x_i^{ini} \oplus y_i^{ini}})$ with K_S:

$\boldsymbol{EDataResponse = encrypt(K_S, DataRes||(x_i^{ini} \oplus y_i^{ini}))}$

6: S_i sends the response to the master: $\boldsymbol{S_i \xrightarrow{public} M : EDataResponse}$
response: $[\boldsymbol{i, FC, EDataResponse, CRC}]$

7: M receives the response: $\boldsymbol{M \xleftarrow{public} S_i : EDataResponse}$

8: M decrypts the response:

$\boldsymbol{DataRes}\ ||(\boldsymbol{x_i^{ini} \oplus y_i^{ini}}) = \boldsymbol{decrypt(K_S, EDataResponse)}$

9: M makes the checks:

$\boldsymbol{if}\ \boldsymbol{x_i^{ini} \oplus (x_i^{ini} \oplus y_i^{ini}) = y_i^{ini}\ then}$

$\boldsymbol{if}$ $DataRes$ is correct $\boldsymbol{then}$ M processes the response

$\boldsymbol{else}$ M goes back to initial mode

$\boldsymbol{end\ if}$

$\boldsymbol{else}$ M goes back to initial mode

$\boldsymbol{end\ if}$

4.2.3.2.2 Communication with sensor slaves

The communication method between sensors and masters are similar to the communication presented in the previous section. However, the data field of the messages contains different information. In the case of actuator-master communication, the data field contains some additional information to the defined

function code. Sensors provide more complex data. To extend the level of data integrity, the entropy of the system is increased.

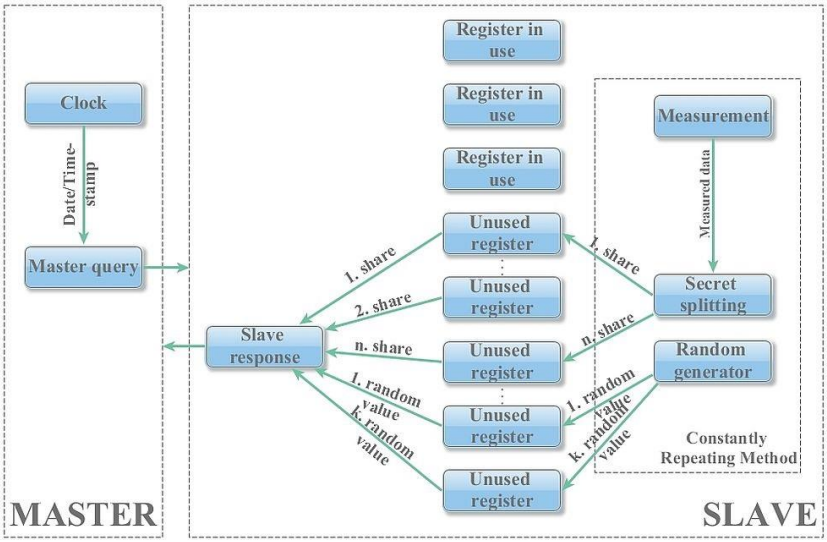

Figure 36. Secure Modbus RTU communication on the Sensor Slave side

The sensor device displayed in Figure 36. is an end node of a SCADA system. In predefined intervals, that slave device measures given characteristics of the system and then saves the gathered data in declared registers. If the master sends a request, then the slave responds to it with the measured data included. In this solution, the Modbus RTU protocol is modified, to be able to divide the measured data between the unused register of the slave device, instantly after the measurement is made. The unused registers are filled with data generated from the measurement by functions of the secret key and the challenge values. The number of the part – divided into – and the secret key and the challenges define the position of every secret part. Secret sharing is realized in this way and resulted in several so-called imprints. Later all of these imprints are required to recreate the original measured data – as opposed to the traditional secret sharing method – because the creation of these imprints only makes the access more difficult to the secret, but not impossible. The more registers are participating in the decryption method, the more grows the entropy of the system, which in turn makes it more difficult for an attacker to gather relevant data. Keep in mind that if the number of imprints is the same as the number of unused registers the decrypting of the encrypted data become trivial, but the message cannot be consisted only of these imprints, so after secret-sharing, the free registers filled up with randomized data seemingly similar to the imprints. The steps of the generation of content of the data field of the request are given below.

1: S_i measures a data: $\boldsymbol{MesuredData}$

2: S_i makes the following calculations: $\boldsymbol{numberOfShares = K_S \, mod \, C_i}$

3: S_i calculates the position of the registers to put shares inside:

$\boldsymbol{for}\ m = 0\ \boldsymbol{to}\ m < numberOfShare\ \boldsymbol{do}$

$\boldsymbol{numShareReg_m} =$

$(\boldsymbol{numberOfShares + m}) \boldsymbol{mod}\ (\boldsymbol{numOfUUReg + 1})$

$\boldsymbol{a_i^{com}} = \boldsymbol{random}(\ \ )$

$\boldsymbol{x_m^{com}} = \boldsymbol{random}(\ \ )$

$\boldsymbol{end\ for}$

4: S_i calculates the position of the registers to put random values inside:

$\boldsymbol{for}\ m = numberOfShares + 1\ \boldsymbol{to}\ m < numOfUUReg\ \boldsymbol{do}$

$\boldsymbol{numRandomReg_m} =$

$(\boldsymbol{numberOfShares + m}) \boldsymbol{mod}\ (\boldsymbol{numOfUUReg + 1})$

$\boldsymbol{end\ for}$

5: S_i constructs the random polynomial:

$\boldsymbol{P_{com}(x^{com}) = MeasuredData} +$

$\boldsymbol{a_1^{com} x^{com} + a_2^{com} x^{com2} \ldots + a^{com}_{numberOfShares} x^{com numberOfShares}}$

6: S_i puts share into the right register

$\boldsymbol{for}\ m = 0\ \boldsymbol{to}\ m < numOfShare\ \boldsymbol{do}$

$\boldsymbol{contentReg_{numShareReg_m}} =$

$(\boldsymbol{x_m^{com}} || \boldsymbol{P_{com}(x_m^{com})}) \oplus \boldsymbol{C_i} || \boldsymbol{00 \ldots 0}$

$\boldsymbol{end\ for}$

7: S_i puts random values into the right register

$\boldsymbol{for}\ m = numOfShares + 1\ \boldsymbol{to}\ m < numOfUUReg\ \boldsymbol{do}$

$\boldsymbol{contentReg_{numRandomReg_m}} = (\boldsymbol{random}(\ \ ) \oplus \boldsymbol{C_i} | \boldsymbol{00 \ldots 0}$

$\boldsymbol{end\ for}$

8: S_i constructs the data part of the response:

$\boldsymbol{contentReg_1 || contentReg_2 || contentReg_{numOfUUReg}}$

4.2.3.3 TEST SYSTEM

A test system was designed and constructed in the Department of Electrical Engineering and Mechatronics at the University of Debrecen. The system consists of a PC, a MOXA converter, a digital oscilloscope a PT1008 data acquisition device, and eight PTC thermoresistores. The PC serves as the master of the system, a simple Modbus client software was running and logging the gathered data to a comma-separated text file in it. The converter and the digital oscilloscope is planted into the system for providing the understanding between the HMI and the actuator device, which is the PT1008. The actuator is connected to eight thermoresistores which are able to measure temperature and serves as servers in the system. The microcontroller, driving the actuator device, is an Atmel ATxmega 128a, and the firmware is implemented in Basic. The secure protocol was also implemented in

Basic. The eight PTC thermoresistors supplied the data to be encrypted. These thermic sensors measured the temperature of the MSc laboratory of the department, where the system was installed. The physical layer of the network was RS485 2-wire bus, the end nodes connected with the master via a Moxa Serial-to-USB converter device. The architecture of the system is displayed in Figure 37.

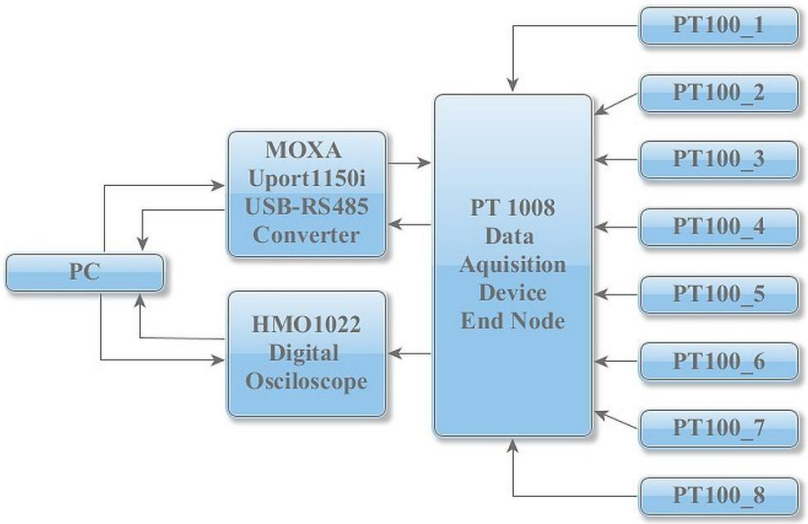

Figure 37. Structure of the test system

The running time of a single message exchange with our secure protocol is completed five times longer than with the original Modbus RTU protocol. The overall performance of the device has not been reduced, all of its functionality has remained the same; thus it can be stated that the protocol is able to be implemented in field devices with lower performance. The results of the test case are the following. Time of generation of the response on the slave side – initialization is not included – has increased with 400%. The time needed for transmitting a response to the master has grown by 23%. Over the testing period, the number of dropped messages was 1:15 000.

4 SUMMARY AND THESIS POINTS

In the present dissertation, the vulnerabilities of two different communication methods were examined. The first one is the communication between the GPS satellites and receivers, and the second one is the communication between field devices in a Modbus RTU based SCADA system. The complete technical description is presented for both of the underlying systems. Example applications are given and described to show the broad applicability of both systems and to draw attention to how crucial it is to secure these communication flows. Vulnerabilities and security breaches are revealed by comprehensively studying the related scientific literature, and available solutions are presented for both of the examined systems. For the problem of securing the communication between GPS satellites and receivers, two solutions are made and given, with different level of provided security. For securing the communication between field devices in a Modbus RTU based SCADA system, one solution is designed and presented. Correctness check is implemented for the secure protocol created for GPS location calculation, and a test system is built to check the running parameters of the secure Modbus RTU communication.

5.1 FIRST THESIS POINT

The 3rd Section of the dissertation is devoted to revealing the security problems of different GNSS systems, especially the GPS, and overviewing the existing solutions for the disclosed security breaches by comprehensively examining the related scientific literature. Additionally, to give solutions for the revealed security problems and provide a way to authenticate location information from a cryptographical point of view.

5.1.1 REVEALING THE WEAKNESSES OF THE GLOBAL NAVIGATION SATELLITE

Even though the number of GNSS based services, especially the GPS based ones, has been increased enormously in the last decades, only a small part of the security issues of such systems got enough attention in the related scientific literature. Through these years the GNSS technology has gained a considerable portion in the civil, scientific, industrial, governmental, or military sectors, and the list is not complete at all. In Section 3.2.1 and 3.2.2 of the dissertation, the result of a comprehensive examination about the vulnerabilities of the American GNSS, i.e. the GPS is given. In the time of our research, it was the most popular, commonly used and stable GNSS system. Only a few information was available at that time in the related scientific literature about the cryptographic reliability of the entire GPS. On the other hand, this approach is an essential part of the system, because there exist several GPS based service which has to provide cryptographically authenticated data, so it would be necessary to make location determination cryptographically

secure and reliable. The research revealed that two types of threats could be distinguished, the ones that are attacking the geodetic authenticity of the system and the ones that are targeting the cryptographic authenticity of it. Geodetic authenticity can be threatened by intentional and malicious attackers through jamming and spoofing. Cryptographic authenticity is vulnerable because there can be no secure channel provided between the participants, so confidentiality, authenticity, data integrity, and freshness are not guaranteed parameters of the system. Geodetic weaknesses like jamming and spoofing are exhaustively examined in the related literature, and several solutions are performed. At the beginning of the 2000s, because of the widespread usage of GPS, several researchers started to design solutions for the cryptographical authentication shortcomings of the system too. A lot of possible methods were made, but it is important to mention, that all these solutions require fundamental changes in the structure of the GPS signals, so without the evolution of the system cannot be applied. **As a summary of my research, it can be stated that a solution which provides an authenticated location in a cryptographical sense and does not require fundamental changes in the structure of the GPS signal can be a niche (can be found in 1., 2.).**

5.1.2 DESING AND REALISATION OF A „LOCATION-STAMPING" PROTOCOL ON THE SOFTWARE LEVEL

To solve the problem revealed in the first thesis point and provide cryptographically authentic location information without redesigning the GPS signal, two „location-stamping" protocols were made by my co-author and me in 2013 (can be found in 2.). **The developed protocols can provide authentic location and time information for any device which has a GPS receiver built in it. The first solution provides safety with the help of a software component, a trusted third party, and a mobile phone service provider**. The aim of the above lower-safety solution, a.k.a the protocol built in the software level of the mobile device, is to provide a solution that is less hardware dependent. It can be done because it is much easier to access to the calculated GPS coordinates than to the Navigational Message broadcasted by the satellites. To authenticate the GPS information of the Mobile Device, a trusted third party is necessary, that is able to generate the location information of the Mobile Device. Then, the location information of the trusted third party can be compared to the ones received by the Mobile Device. One option to get suitable location information for comparison is to use mobile phone services, which provide cell information. However, cell information is not always accurate enough to specify the location of the Mobile Device. It is a proper solution if the mobile network coverage is broad enough. In the above situation, the mobile phone service has independent information on the location of the Mobile Device, which can be compared to the data that the Mobile Device has sent to the trusted third party.

5.1.3 DESING AND REALISATION OF A „LOCATION-STAMPING" PROTOCOL ON THE HARDWARE LEVEL

The second solution provides higher-safety with the help of a software component and a trusted third party. **In this higher-safety solution, the main goal is to receive and preserve the content of the Navigational Message, transmitted by the satellites, before anybody could modify it (can be found in [2])** The Authentic Software is intended to be built in a very deep and hidden layer of the Mobile Device, namely in its driver level. The protocol, described in detail in Section 3.2.3.2 of the dissertation, has three important participants, which are the satellites of the Global Positioning System, the Mobile Device, and the Certification Authority. The designed higher-level protocol is built in a portable electronic device, which has been patented by the University of Debrecen in the US with the name "Portable electronic device, system, and method for authenticating a document associated with a geographical location" (can be found in 3.). Correctness proof was made with the Applied-π method, and the ProVerif software tool based on the events declared in the processes. The declared events label the steps of the protocol. The correspondence between the earlier declared events are investigated during correctness proof. The authenticity of the participants and the data integrity is proved by checking the correspondence assertions and injective correspondence for events of the subprocesses of the participants (can be found in 7.).

5.2 SECOND THESIS POINT

The 4th Section of the dissertation is devoted to uncovering the vulnerabilities of the Modbus RTU industrial communication protocol, which is one of the basic building blocks of such SCADA systems mentioned in Section 4.1.1.3. Additionally, to overview the existing solutions for the revealed vulnerabilities by comprehensively studying the related scientific literature, and to give a solution for the uncovered weaknesses from a cryptographical point of view.

5.2.1 REVEALING THE WEAKNESSES OF THE MODBUS PROTOCOL

Drinking fresh water, turning the lights on, travelling by tram, calling our family, or getting a medical treatment are usual activities. However, the underlying Critical Infrastructures which are maintained and controlled by the so-called "Supervisory Control and Data Acquisition systems" were always targeted by different types of attacks. During the last decades, because of the fast spread of internet-based services and continuous technical development, Critical Infrastructures become more vulnerable than ever, and cyberattacks happen more frequently. Consequences of such cyberattacks can be not only financial loss, or devastated reputation, but it can be extremely detrimental in health and safety. SCADA systems are often using outdated communication protocols, like the Modbus over Serial Line

as an underlying network so that an update would be required. However, full reconstruction and innovative changes in legacy SCADA systems have a high cost, and it is not always possible to carry out. In this dissertation, the focus is on the Modbus protocol, especially the Modbus over Serial Line, a.k.a the Modbus RTU. The Modbus was originally designed and made by Modicon in 1979. The Modbus RTU protocol does not include any security considerations in default. During the design process, the required features of the Modbus protocol were reliability, speed, and accessibility, but the security of the used channel, thus authentication of the participants, confidentiality, data integrity, and freshness of the messages were left out, so not provided. **Our research – presented in Section 4.2.1 and 4.2.2 of the dissertation – addressed to find the vulnerabilities of a Modbus RTU based SCADA system by comprehensively reviewing the related scientific literature, and applying the Attack Tree method for summarizing and classifying the discovered security breaches (can be found in 4., 5., 6.).** Liabilities of Modbus RTU, revealed by us, are the lack of confidentiality, integrity, authentication, the sensibility for the Man in the middle attack, for Denial of Service Attack, the possibility of interception, interruption, modification, and fabrication. In the available literature, a proper solution for the abovementioned problems cannot be found in the case of Modbus RTU. Thus, the aims to be reached are to prevent the messages from eavesdropping, to authenticate the slaves and the master of the network, and to provide data integrity and freshness of the messages sent via Modbus RTU.

5.2.2 Desing and Realisation of a Secure Modbus RTU protocol

In Section 4.2.3 of the dissertation, our solution for the previously revealed vulnerabilities of the Modbus RTU protocol is presented. The solution is taken advantage of the fact, that the Modbus RTU based communication not always use its whole standardized message length and the messages have a relatively low update frequency. Usually, the Master Terminal Units and field devices have a built-in AES engine to encrypt the standard request and response before sending it. This engine is integrated by the device manufacturers and has no connection with the applied industrial communication protocol. However, the AES engine alone, with encrypting the messages before sent through the channel is not enough to solve all the revealed problems of the Modbus RTU protocol.

Our solution provides authentication of the participants to each other, and additionally data integrity, freshness, and confidentiality of the messages. A test system was designed and constructed in the Department of Electrical Engineering and Mechatronics at the University of Debrecen to check the applicability of the new solution (can be found in 8.). The running time of a single message exchange, applying our secure protocol, is five times longer than with the original Modbus RTU protocol. The overall performance of the device has not been reduced; all of its

functionality has remained the same. Thus it can be stated that the protocol can be implemented in field devices also with low computational capacity. The results gained by applying our test system are the following: Time of generating a response on the slave side – initialization is not included – has increased with 400%. The time needed to transmit a response to the master has grown by 23%. Over the testing period, the rate of dropped messages was 1:15 000.

5 ÖSSZEFOGLALÁS ÉS TÉZISPONTOK

Kutatómunkám során két különböző kommunikációs folyamat sebezhetőségét vizsgáltam kriptográfiai szempontból. Az első kommunikációs folyamat a GPS (Global Positioning System) rendszerben, a GPS műholdak és a GPS vevőkészülékek között valósul meg. Ebben az esetben a célom a vevőkészülék GPS koordinátákkal adott földrajzi helyének kriptográfiai hitelesítése volt. A második kommunikációs folyamat a SCADA (Supervisory Control And Data Acquisition) rendszerekben, a rendszer terepi eszközök között valósul meg a Modbus RTU (Remote Terminal Unit) ipari kommunikációs protokoll segítségével. Itt a célom a terepi eszközök közötti biztonságos kommunikációs csatorna létrehozása volt, vagyis a résztvevők hitelesítésének, valamint az üzenetek bizalmasságának, integritásának és frissességének biztosítása volt. Mindkét esetben olyan megoldást kerestem, amely a kommunikáció alapjául szolgáló rendszerek fizikai struktúráját és adatátviteli keretrendszerét nem érinti. Dolgozatomban a vizsgált rendszerek működési alapelveit részletesen tárgyalom, majd példákkal szemléltettem a rendszereken alapuló különböző szolgáltatásokat, így megmutatva az általam vizsgált kommunikációs folyamatok kriptográfiai biztonságának alapvető fontosságát. A GPS-műholdak és vevők közötti kommunikáció biztonsági problémájának kiküszöbölésére két megoldást (protokollt) adtunk szerzőtársammal, amelyek különböző szintű kriptográfiai és/ vagy geodéziai védelmet biztosítanak. Ezek közül az egyik protokoll helyességének bizonyítása is megtörtént. A terepi eszközök közötti kommunikáció kriptográfiai védelmére egy megoldás adtunk, illetve megépült egy SCADA rendszer az elkészült protokoll futási paramétereinek és használhatóságának tesztelésére.

6.1 ELSŐ TÉZISPONT

A disszertációm harmadik fejezete a különböző GNSS rendszerekkel, ezek közül elsősorban a GPS rendszerrel kapcsolatos eredményeimet ismerteti. Egy átfogó irodalomkutatás és feldolgozás során feltártam a GPS rendszerek biztonsági problémáit mind geodéziai, mind kriptográfiai szempontból, valamint áttekintettem és elemeztem a megtalált biztonsági hiányosságok kiküszöbölésére már létező megoldásokat. Szintén ebben a fejezetben mutatok be két, a szerzőtársammal közösen készített „Helyszín-bélyegző” protokollt, amelyekkel a hiányzó biztonsági paraméterek okozta problémák kiküszöbölhetőek. **Ezekkel a protokollokkal a GPS rendszer által szolgáltatott helyadatok kriptográfiai szempontból hitelesen és letagadhatatlanul hozzárendelhetőek egy adott vevőkészülékhez**. **Az első tézispont a GPS rendszer hiányosságainak feltárását, valamint a két „Helyszín-**

bélyegző" protokoll (az egyik szoftver a másik hardver szintre integrált) tervezését és megvalósítását tartalmazza.

6.1.1 A GPS RENDSZER HIÁNYOSSÁGAINAK FELTÁRÁSA

A GNSS-alapú szolgáltatások gyakoriak polgári, tudományos, ipari, kormányzati vagy katonai alkalmazásokban, számuk az elmúlt néhány évtizedben rendkívüli mértékben nőtt. Ennek ellenére a kapcsolódó szakirodalom a GNSS rendszerek biztonsági hiányosságainak csak egy szűk szegmensével foglalkozik. A 3.2.1. és 3.2.2. fejezetekben az amerikai GNSS, azaz a GPS rendszer hiányosságait feltáró vizsgálataimat, és azok eredményét ismertetem.

Kutatásunk idején a GPS volt a legnépszerűbb, leggyakrabban használt, legnagyobb lefedettségű és a legstabilabban működő GNSS rendszer, mégis csak igen kevés információ állt rendelkezésre a GPS rendszer kriptográfiai biztonságáról. Pedig számos olyan GPS-alapú szolgáltatás létezett és létezik ma is, ahol elengedhetetlen, hogy a helymeghatározás kriptográfiai szempontból is biztonságos legyen. Dolgozatomban azon szolgáltatásokkal foglalkoztam, ahol a GPS koordinátáknak valamely jogi eljárásban bizonyító ereje van.

A GPS rendszer két szempontból is sérülékeny: geodéziai és kriptográfiai hitelessége is sérülhet egy rosszindulatú támadás során. A geodéziai szempontból hiteles helymeghatározást veszélyeztető támadások két típusa a „Zavarás" és a „Megtévesztés". „Zavarás" esetén a támadó – a GPS jel elfedésével – a teljes szolgáltatást elérhetetlenné teszi a GPS vevő számára, míg „Megtévesztés" esetén a valóstól eltérő helyadatokkal téveszti meg azt. Ezen támadások azért kivitelezhetőek, mert a műholdak által sugárzott jel a földfelszín közelébe érve már jelentősen gyengül. Az fenti támadástípusok elhárítására több módszer is ismert, amelyeket a vonatkozó szakirodalom részletesen tárgyal.

Kriptográfiai szempontból más a helyzet, mivel a műholdak és a vevők közötti kommunikációs csatorna nem rendelkezik az alapvető biztonsági jellemzőkkel: nevezetesen az üzenetek bizalmassága, integritása és frissessége, valamint a résztvevők hitelessége sem garantált. Emellett a vevőkészülékek sem tekinthetők kriptográfiai szempontból megbízhatónak, mivel az általuk fogadott adatok, vagy az általuk számított koordináták nincsenek megfelelően védve a rosszindulatú módosításoktól. A 2000-es évek eleje óta folynak kutatások a rendszer kriptográfiai hiányosságainak pótlására. Több módszer is született, de ezek kivétel nélkül alapvető változtatásokat igényelnek a GPS-jel szerkezetében, tehát a rendszer átalakítása nélkül nem alkalmazhatóak.

Összefoglalva elmondható, hogy olyan eljárás, amely nem igényel alapvető változtatásokat a GPS-jel szerkezetében, és amely kriptográfiai értelemben

hitelesíti a helymeghatározást, azaz biztosítja az üzenetek bizalmasságát, integritását, frissességét, valamint a résztvevők hitelességét, emellett a vevőkészülékekhez hitelesen és letagadhatatlanul hozzárendeli az általuk számolt GPS koordináták által adott helyet és időt, a vizsgálataim előtt nem állt rendelkezésre. (lásd 1., 2.)

6.1.2 SZOFTVER SZINTRE INTEGRÁLT „HELYSZÍN-BÉLYEGZŐ" PROTOKOLL TERVEZÉSE ÉS MEGVALÓSÍTÁSA

A 2.1.1 fejezetben elmondottak szerint, vizsgálataim kezdetén nem állt rendelkezésre olyan eljárás, amely egyrészt a GPS jel szerkezetének változtatása nélkül képes biztosítani azon biztonsági kritériumokat, melyek a GPS műholdak és a GPS vevők közötti biztonságos csatorna kiépítéséhez szükségesek, másrészt garantálja a vevők által vett és számított adatok kriptográfiai védelmét. Az általunk 2013-ben készített, szoftver szintre integrált „helyszín-bélyegző" protokoll (lásd 2.) a második problémára nyújt megoldást. **A szoftver szintre integrált „Helyszín-bélyegző" protokoll egy olyan szoftver, amely egy megbízható harmadik fél és egy mobilszolgáltató segítségével garantálja a vevő által kezelt adatok kriptográfiai védelmét, emellett geodéziai hitelességet is biztosít bármely beépített GPS vevővel rendelkező eszköz számára.** A megoldás hardverfüggetlen, mivel a hitelesítést egy a vevőre telepített szoftverrel valósítjuk meg. A módszer alkalmazása során a koordináták számítása és aláírása a mobileszköz szoftver szintjén történik. A hiteles „Helyszín-bélyeghez" szükséges egy, a GPS rendszertől független rendszer helyadatainak összehasonlítása az eredetivel. Az eljárás során használt digitális aláírás hitelesíti, hogy a számított GPS koordinátákat a vevőkészülék küldte a hitelesítő szervezetnek, és annak tartalma az átvitel során nem változott. A mobilszolgáltató által nyújtott helyinformációk összevetése a vevőkészülék által aláírt GPS koordinátákkal a helyadatok geodézia hitelességét hivatott igazolni, valamint megnöveli a kriptográfiai hitelesség szintjét is. A számított GPS koordináták frissességét a protokoll egy véletlen értékkel és a GPS navigációs üzenetéből nyert időinformációval biztosítja.

6.1.3 HARDVER SZINTRE INTEGRÁLT „HELYSZÍN-BÉLYEGZŐ" PROTOKOLL TERVEZÉSE ÉS MEGVALÓSÍTÁSA

Az általunk tervezett és kivitelezett második protokoll (lásd 2.), magasabb szintű kriptográfiai védelmet biztosít, mint az első, de ennek érdekében a vevőkészülékekben hardver szintű módosításokat kíván. A dolgozat 3.2.3.2. fejezetében részletesen ismertetett protokoll résztvevői a GPS rendszer, a vevőkészülék és egy megbízható harmadik fél, jelen esetben egy hitelesítő szervezet. **Elsődleges cél a folyamatban a műholdaktól érkező navigációs üzenet vétele és**

rögzítése úgy, hogy a vevőkészülékben a rosszindulatú módosítások ne történhessenek meg. A módszer megvalósítása során a koordináták számítása és aláírása a mobileszköz hardver szintjére integrált. A protokoll esetében a digitális aláírás, valamint a biztonságos hardverelem hitelesíti, hogy az adott mobileszköz fogadta az aláírt adatokat a műholdtól, továbbá ez az eszköz számította a GPS koordinátákat is. A számított GPS koordináták sértetlenségét is a digitális aláírás garantálja. A számított GPS koordináták frissességét a protokoll egy véletlen értékkel és a GPS navigációs üzenetéből nyert időinformációval biztosítja. A protokoll egy, a Debreceni Egyetem által *"Portable electronic device, system, and method for authenticating a document associated with a geographical location"* (lásd 3.) névvel szabadalmaztatott eljárás alapját képezi. Ezen módszer esetében a ProVerif automatikus helyesség bizonyító szoftver alkalmazásával igazoltam a protokoll által szolgáltatott biztonsági paraméterek közül kettőt (lásd 7.).

6.2 MÁSODIK TÉZISPONT

A disszertációm negyedik fejezete a Modbus RTU ipari kommunikációs protokollt használó SCADA rendszerekkel kapcsolatos kutatási eredményeimet tartalmazza. Egy átfogó irodalomkutatás és feldolgozás során feltártam a Modbus RTU protokoll biztonsági hiányosságait, és az azok kiküszöbölésére már meglévő eljárásokat áttekintettem és elemeztem. Ebben a fejezetben ismertetem a kriptográfiailag biztonságos átviteli csatorna megvalósítására kidolgozott "Biztonságos Modbus RTU" protokollt is. **Az általam elkészített protokoll a Modbus RTU adatátviteli keretének és fizikai struktúrájának változtatása nélkül képes biztosítani a hiányzó alapvető biztonsági kritériumokat. A dolgozat második tézispontja a Modbus RTU protokoll hiányosságainak az "Attack-tree" módszerrel történő megvalósítását és a "Biztonságos Modbus RTU" protokoll tervezését, implementálását és tesztelését tartalmazza.**

6.2.1 A MODBUS RTU HIÁNYOSSÁGAINAK FELTÁRÁSA

Meginni egy pohár tiszta vizet, felkapcsolni a villanyt, felhívni a családunkat, vagy igénybe venni valamilyen orvosi kezelést mind hétköznapi tevékenységek, melyek biztosításáért ún. kritikus infrastruktúrák felelősek. A kritikus infrastruktúrákhoz tartoznak a vízelosztó-, az áramszolgáltató és a telekommunikációs hálózatok, de az egészségügyi ellátórendszer létesítményei is. Az említett rendszerek esetében általában az irányítást, az adatgyűjtést és a rendszerek felügyelete SCADA rendszerek segítségével valósul meg. A SCADA rendszerek működése különböző kommunikációs protokollokra és számtalan terepi eszközre támaszkodik. Azonban a vezetéknélküli kommunikáció térnyerése, valamint a SCADA hálózatok online jellege miatt, a kritikus infrastruktúrák sebezhetősége jelentősen megnőtt az elmúlt évtizedekben. A fenti rendszereket nap, mint nap érik kiber támadások, melyek következményei

nem csupán pénzügyi veszteség, vagy megromlott vállalati hírnév lehetnek, hanem az állampolgárok életének veszélyeztetése is. A nagy hálózatok gyakran régi, elavult kommunikációs protokollokra épülnek, mint a Modbus RTU ipari kommunikációs protokoll, de a teljes felújítás sokszor nem kivitelezhető, vagy csak nagyon magas költség mellett. A dolgozatom fókuszában a Modbus RTU ipari kommunikációs protokoll áll, amely a Modbus családdal együtt 1979 óta szabvány. A Modbus protokoll tervezése során a biztonságos kommunikáció, mint tervezési cél kimaradt, a megbízható működésre és a megfelelő sebességre fektettek csak hangsúlyt. A dolgozatom 4.2.1. és 4.2.2. fejezetében bemutatott kutatásaim célja a Modbus RTU alapú SCADA rendszerek sebezhetőségének vizsgálata a kapcsolódó szakirodalom átfogó áttekintésével és elemzésével, továbbá az „Attack-Tree" módszer alkalmazásával. (lásd 4., 5., 6.). **A kutatás eredményeként megtalált biztonsági hiányosságok egyrészről a résztvevők hitelesítésének, valamint az üzenetek bizalmasságának, integritásának és frissességének hiánya az adatátvitelre használt csatornán, másrészről a protokoll érzékenysége a MITM (Man in the Middle) és a DoS (Denial of Service) típusú támadásokra. Harmadrészt, hogy fennáll a csatornán haladó üzenetek lehallgatásának, módosításának, fabrikálásának és az üzenettovábbítás megszakításának lehetősége. A vonatkozó szakirodalomban, a vizsgálataim kezdetén, a fenti problémákra nem volt létező megoldás.**

6.2.2 Biztonságos Modbus RTU kommunikáció tervezése és megvalósítása

A dolgozat 4.2.3-as fejezetében a Modbus RTU ipari kommunikációs protokoll biztonsági hiányosságainak megoldására kifejlesztett "Biztonságos Modbus RTU" protokollt mutatom be. A javasolt módszer azon alapul, hogy a Modbus RTU protokoll az adatátvitel során nem használja ki maximálisan a lehetséges üzenethosszt, és az üzenetváltás gyakorisága relatíve alacsony. Jellemzően a gyakorlatban a SCADA rendszerekben mind az MTU (Master Terminal Unit), mind a terepi eszközök rendelkeznek egy beépített AES (Advanced Encryption Standard) titkosítóval, mely a szabványos kéréseket és válaszokat képes titkosítani a csatornán. Ez az ún. AES motor az eszközök sajátja, egy a gyártók által beépített szolgáltatás, mely független az alkalmazott ipari kommunikációs protokolltól. Meg kell jegyezni, hogy habár a beépített AES titkosító elősegíti a biztonságos átviteli csatorna kiépítését, de önmagában nem oldja meg a fentebb bemutatott biztonsági hiányosságokat.

Az általunk tervezett és megvalósított megoldás a résztvevők hitelesítését, az üzenetek adatintegritását, frissességét és bizalmasságát garantálja. A közös titkos kulcs és a kihívás értékek hitelesítik a Mestert a Szolga felé és fordítva. Az üzenetek bizalmassága egy beépített AES titkosítóval biztosított. Az adatok integritását a

Szenzor típusú Szolgák esetén egy ún. „összekeverés” nevű eljárás biztosítja, amely titokmegosztáson alapul (lásd 8.).

A „Biztonságos Modbus RTU” protokoll gyakorlati alkalmazhatóságának ellenőrzésére a Debreceni Egyetem Műszaki Karának Villamosmérnöki és Mechatronikai Tanszékén megterveztünk és kiviteleztünk egy teszt rendszert Az alábbiakat tapasztaltuk: Egy egyszeri üzenetváltás sebessége ötször nagyobb a javasolt biztonságos módszer alkalmazása esetén, mint az eredeti protokoll használatával. A terepi eszközök teljesítménye nem csökkent, minden funkciójuk hibátlanul, a megszokott módon működött. A szolga oldalon történő válasz üzenet generálásának ideje 400%-kal nőtt, ebbe a kommunikáció inicializáló lépései nem kerültek beszámításba. A mester felé történő üzenettovábbításhoz szükséges idő hossza 23%-kal nőtt. A tesztelési periódus során az elutasított üzenetek aránya 1:15000 volt. A teszt alapján kijelenthetjük, hogy a módszer kis számítási kapacitással rendelkező terepi eszközök esetén is alkalmazható.

REFERENCES

[1] News Desk „South Korea to track quarantine violators through tracking wristbands," Geospatial Wolrd, 2020
[Online]. https://www.geospatialworld.net/news/south-korea-to-track-quarantine-violators-through-tracking-wristbands/.

[2] Turzó, Á. P. „A koronavírust is megfékezheti az app, amit egy magyar cég fejleszt," Portfolió, 2020
[Online]. https://www.portfolio.hu/uzlet/20200320/a-koronavirust-is-megfekezheti-az-app-amit-egy-magyar-ceg-fejleszt-420965.

[3] Shannon, C. E. (1948). A mathematical theory of communication. Bell system technical journal, 27(3), 379-423.

[4] Shannon, C. E., & Weaver, W. (1949). The mathematical theory of communication, 117 pp. Urbana: University of Illinois Press.

[5] Dolev, D., & Yao, A. (1983). On the security of public key protocols. IEEE Transactions on information theory, 29(2), 198-208.

[6] Gasmi, Y., Sadeghi, A. R., Stewin, P., Unger, M., & Asokan, N. (2007, November). Beyond secure channels. In Proceedings of the 2007 ACM workshop on Scalable trusted computing (pp. 30-40).

[7] Williams, N. (2007). On the use of channel bindings to secure channels. RFC 5056, November.

[8] Maurer, U. M., & Schmid, P. E. (1994, November). A calculus for secure channel establishment in open networks. In European Symposium on Research in Computer Security (pp. 173-192). Springer, Berlin, Heidelberg.

[9] Basin, D., Radomirovic, S., & Schläepfer, M. (2015, July). A complete characterization of secure human-server communication. In 2015 IEEE 28th Computer Security Foundations Symposium (pp. 199-213). IEEE.

[10] Clarkson, System Security Course: Secure Channel 2019, Cornell University, 2019
[Online]. www.cs.cornell.edu/courses/cs5430/2016sp/l/09-channel/lec.pdf

[11] Maurer, U., Rüedlinger, A., & Tackmann, B. (2012, March). Confidentiality and integrity: A constructive perspective. In Theory of Cryptography Conference (pp. 209-229). Springer, Berlin, Heidelberg.

[12] Menezes, A. J., Katz, J., Van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of applied cryptography. CRC press.

[13] Blanchet, B. (2016). Modeling and verifying security protocols with the applied pi calculus and ProVerif. Foundations and Trends® in Privacy and Security, 1(1-2), 1-135.

[14] Blanchet, B. (2002, September). From secrecy to authenticity in security protocols. In International Static Analysis Symposium (pp. 342-359). Springer, Berlin, Heidelberg.

[15] Forouzan, B. A., & Mukhopadhyay, D. Cryptography and network security (Sie). McGraw-Hill Education, Southern Illinois University,2011 [Online]. www.cs.siue.edu/~tgamage/S18/CS490/L/WK12.pdf.

[16] ISO/IEC 9798-1:2010 Information technology — Security techniques — Entity authentication. International Organization for Standardization.

[17] Martin, K. M. (2012). Everyday cryptography. The Australian Mathematical Society, 231(6).

[18] Diffie, W., Van Oorschot, P. C., & Wiener, M. J. (1992). Authentication and authenticated key exchanges. Designs, Codes and cryptography, 2(2), 107-125.

[19] Harkins,D., Carrol, D. „The Internet Key Exchange (IKE), RFC 2409," IETF Tools, 1998
[Online]. https://tools.ietf.org/html/rfc2409.

[20] Matsumoto, T., Takashima, Y., & Imai, H. (1986). On seeking smart public-key-distribution systems. IEICE TRANSACTIONS (1976-1990), 69(2), 99-106.

[21] Menezes, A. (1997). Some new key agreement protocols providing implicit authentication. In Workshop on Selected Areas in Cryptography, 1997. CRC Press.

[22] Chandra, S., Paira, S., Alam, S. S., & Sanyal, G. (2014, November). A comparative survey of symmetric and asymmetric key cryptography. In

2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE) (pp. 83-93). IEEE.

[23] Tripathi, R., & Agrawal, S. (2014). Comparative study of symmetric and asymmetric cryptography techniques. International Journal of Advance Foundation and Research in Computer (IJAFRC), 1(6), 68-76.

[24] Of, Triple Data Encryption Algorithm Modes. "Operation, X9. 52–1998, Accredited Standards Committee X9." American National Standards Institute (1998).

[25] Daemen, J., & Rijmen, V. (1999). AES proposal: Rijndael.

[26] Schneier, B. (1993, December). Description of a new variable-length key, 64-bit block cipher (Blowfish). In International Workshop on Fast Software Encryption (pp. 191-204). Springer, Berlin, Heidelberg.

[27] Schneier, B., Kelsey, J., Whiting, D., Wagner, D., Hall, C., & Ferguson, N. (1998). Twofish: A 128-bit block cipher. NIST AES Proposal, 15(1), 23-91.

[28] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

[29] Rabin, M. O. (1979). Digitalized signatures and public-key functions as intractable as factorization (No. MIT/LCS/TR-212). Massachusetts Inst of Tech Cambridge Lab for Computer Science.

[30] ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory, 31(4), 469-472.

[31] Koblitz, N. (1987). Elliptic curve cryptosystems. Mathematics of computation, 48(177), 203-209.

[32] Miller, V. S. (1985, August). Use of elliptic curves in cryptography. In Conference on the theory and application of cryptographic techniques (pp. 417-426). Springer, Berlin, Heidelberg.

[33] Menezes, A. J., & Vanstone, S. A. (1993). Elliptic curve cryptosystems and their implementation. Journal of Cryptology, 6(4), 209-224.

[34] ANSI, X. (1999). 62: public key cryptography for the financial services industry: the elliptic curve digital signature algorithm (ecdsa). Am. Nat'l Standards Inst.

[35] Smith,N., „An Overview of Public Key Infrastructures (PKI)," Techotopia, 2015.
[Online].
https://www.techotopia.com/index.php/An_Overview_of_Public_Key_Infrastructures_(PKI).

[36] MSDN, „MSDN," Microsoft, 2015.
[Online]. https://docs.microsoft.com/.

[37] Gallagher, P., & Director, A. (1995). Secure hash standard (shs). FIPS PUB, 180, 183.

[38] Zheng, T., Radhakrishnan, S., & Sarangan, V. (2005, April). PMAC: an adaptive energy-efficient MAC protocol for wireless sensor networks. In 19th IEEE International Parallel and Distributed Processing Symposium (pp. 8-pp). IEEE.

[39] Sanz Subirana, J., Juan Zornoza, J. M., & Hernández-Pajares, M. (2013). GNSS Data Processing, Volume I: Fundamentals and Algorithms. ESA Communications, ESTEC, Noordwijk, Netherlands, 145-161.

[40] U.S. Department of Homeland Security, "GPS fully operational statement of 1995", Navigation Center of Excellence, 1995
[Online].www.navcen.uscg.gov/?pageName=global

[41] Subirana, J. S., Zornoza, J. J., & Pajares, M. H. (2011). GNSS signal.

[42] Martin, H. National Coordination Office for Space-Based Positioning, Navigation, and Timing, GPS.gov
[Online]. www.gps.gov

[43] Kaplan, E., & Hegarty, C. (2005). Understanding GPS: principles and applications. Artech house.

[44] Jeffrey, C. (2010). An introduction to GNSS: GPS, GLONASS, Galileo and other global navigation satellite systems. NovAtel.

[45] Mai, T., National Aeronautics and Space Administration, NASA, 2014 [Online]. www.nasa.gov/directorates/heo/scan/communications/policy/GPS.html

[46] Rajendran, V. „Location Based Services : Expected Trends and Technological Advancements," Geo awesomeness, 2017. [Online]. https://geoawesomeness.com/expected-trends-technological-advancements-location-based-services/.

[47] „The Noise App," [Online]. www.thenoiseapp.com.

[48] „allGeo," [Online]. www.allgeo.com.

[49] „alibi," [Online]. appfelstrudel.com/id/957636810/alibi.html.

[50] Papadimitratos, P., & Jovanovic, A. (2008, October). Protection and fundamental vulnerability of GNSS. In 2008 IEEE International Workshop on Satellite and Space Communications (pp. 167-171). IEEE.

[51] Papadimitratos, P., & Jovanovic, A. (2008, November). GNSS-based positioning: Attacks and countermeasures. In MILCOM 2008-2008 IEEE Military Communications Conference (pp. 1-7). IEEE.

[52] Ioannides, R. T., Pany, T., & Gibbons, G. (2016). Known vulnerabilities of global navigation satellite systems, status, and potential mitigation techniques. Proceedings of the IEEE, 104(6), 1174-1194.

[53] Schmidt, D., Radke, K., Camtepe, S., Foo, E., & Ren, M. (2016). A survey and analysis of the GNSS spoofing threat and countermeasures. ACM Computing Surveys (CSUR), 48(4), 1-31.

[54] Sathyamoorthy, D. (2013). Global navigation satellite system (GNSS) spoofing: a review of growing risks and mitigation steps. Defence S&T Technical Bulletin, 6(1), 42-61.

[55] Dixon, C., Smith, S., Hart, A., Keast, R., Lithgow, S., Grant, A., ... & Beatty, C. (2013). GNSS Vulnerabilities at sea. Coordinates, IX (11), 37-51.

[56] Snowball, A. (2007). An update on GNSS vulnerability-threats and solutions.

[57] Zhang, W., Hou, H., Li, Q., & Wang, W. (2013, November). Vulnerability analysis of the global navigation satellite systems from the information flow perspective. In 2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering (Vol. 1, pp. 263-267). IEEE.

[58] Ji-Hyon, K., „MEDCALC," Medcalc, 2020 [Online]. www.medcalc.org/manual/accuracy_precision.php.

[59] Dovis, F. (Ed.). (2015). GNSS interference threats and countermeasures. Artech House.

[60] Hu, H., & Wei, N. (2009, December). A study of GPS jamming and anti-jamming. In 2009 2nd international conference on power electronics and intelligent transportation system (PEITS) (Vol. 1, pp. 388-391). IEEE.

[61] Gao, G. X., Sgammini, M., Lu, M., & Kubo, N. (2016). Protecting GNSS receivers from jamming and interference. Proceedings of the IEEE, 104(6), 1327-1338.

[62] Wang, P. (2016). Research on Blanket Jamming to Beidou Navigation Signals Based on BOC Modulation. International Journal of Communications, Network and System Sciences, 9(05), 135.

[63] Hambling, D. (2017). Ships fooled in GPS spoofing attack suggest Russian cyberweapon. New Scientist, 10.

[64] van der Merwe, J. R., Zubizarreta, X., Lukčin, I., Rügamer, A., & Felber, W. (2018, May). Classification of spoofing attack types. In 2018 European Navigation Conference (ENC) (pp. 91-99). IEEE.

[65] Jafarnia-Jahromi, A., Broumandan, A., Nielsen, J., & Lachapelle, G. (2012). GPS vulnerability to spoofing threats and a review of antispoofing techniques. International Journal of Navigation and Observation, 2012.

[66] Rounds, S. „Jamming Protection of GPS Receivers – Part II. Antenna Enhancements," *GPS World 2004 February,* pp. 28-45, 2004.

[67] Ruegamer, A., & Kowalewski, D. (2015). Jamming and Spoofing of GNSS Signals–An Underestimated Risk?!. Proc. Wisdom Ages Challenges Modern World, 17-21.

[68] Joseph, A. „GNSS solutions: Measuring GNSS signal strength," *Inside GNSS (2010/11-12):,* pp. 20-25, 2010.

[69] Jones, M. „Anti-jam technology: Demystifying the CRPA," *GPS World, April 2012,* 2012.

[70] „Jamming Technology," Navcours, 2019 [Online]. http://www.navcours.com/en/tech/tech_0205.asp.

[71] Papadimitratos, P., & Jovanovic, A. (2008, October). Protection and fundamental vulnerability of GNSS. In 2008 IEEE International Workshop on Satellite and Space Communications (pp. 167-171). IEEE.

[72] Margaria, D., Motella, B., Anghileri, M., Floch, J. J., Fernandez-Hernandez, I., & Paonni, M. (2017). Signal structure-based authentication for civil GNSSs: Recent solutions and perspectives. IEEE signal processing magazine, 34(5), 27-37.

[73] Scott, L. (2001, March). Anti-spoofing & authenticated signal architectures for civil navigation systems. In Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003) (pp. 1543-1552).

[74] Wullems, C., Pozzobon, O., & Kubik, K. (2005, July). Signal authentication and integrity schemes for next generation global navigation satellite systems. In Proceedings of the European Navigation Conference GNSS, 2005.

[75] Margaria, D., Motella, B., Anghileri, M., Floch, J. J., Fernandez-Hernandez, I., & Paonni, M. (2017). Signal structure-based authentication for civil GNSSs: Recent solutions and perspectives. IEEE signal processing magazine, 34(5), 27-37.

[76] Fernández-Hernández, I. (2014). GNSS Authentication: Design Parameters and Service Concepts. In Proceedings of the European Navigation Conference.

[77] Hein, G. et.al. „Authenticating GNSS proofs against spoofs part 2.,” *Inside GNSS 2.5 ,* pp. 71-78, 2007.

[78] Cheng, X. J., Xu, J. N., Cao, K. J., & Wang, J. (2009, November). An authenticity verification scheme based on hidden messages for current civilian GPS signals. In 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology (pp. 345-352). IEEE.

[79] Mark, P. „What is navigation message authentication?,” *Inside GNSS (2018/01-02),* pp. 26-31, 2018.

[80] Motella, B., Margaria, D., & Paonni, M. (2018, April). SNAP: An authentication concept for the Galileo open service. In 2018 IEEE/ION Position, Location and Navigation Symposium (PLANS) (pp. 967-977). IEEE.

[81] US Marine Corps, “Marine Artillery Survey Operations”, MCVP 3-16.7

[82] Ádámkó, E., & Pethő, A. (2013). Location-stamp for GPS coordinates. Acta Universitatis Sapientiae, Informatica, 5(1), 63-76.

[83] Kabatnik, M., & Zugenmaier, A. (2001, July). „location-stamp”s for digital signatures: a new service for mobile telephone networks. In International Conference on Networking (pp. 20-30). Springer, Berlin, Heidelberg.

[84] Blanchet, B. (2012, March). Security protocol verification: Symbolic and computational models. In International Conference on Principles of Security and Trust (pp. 3-29). Springer, Berlin, Heidelberg.

[85] Ádámkó, É. (2017). Security analysis of a „Location-stamping” protocol for GPS coordinates. Műszaki és Menedzsment Tudományi Közlemények, 2(2), 1-12.

[86] Meadows, C. A. (1994, November). Formal verification of cryptographic protocols: A survey. In International Conference on the Theory and Application of Cryptology (pp. 133-150). Springer, Berlin, Heidelberg.

[87] Abadi, M., Blanchet, B., & Fournet, C. (2017). The applied pi calculus: Mobile values, new names, and secure communication. Journal of the ACM (JACM), 65(1), 1-41.

[88] Abadi, M., & Fournet, C. (2001). Mobile values, new names, and secure communication. ACM Sigplan Notices, 36(3), 104-115.

[89] Milner, R. (1999). Communicating and mobile systems: the pi calculus. Cambridge university press.

[90] Abadi, M., & Gordon, A. D. (1999). A calculus for cryptographic protocols: The spi calculus. Information and computation, 148(1), 1-70.

[91] Abadi, M., & Blanchet, B. (2003, June). Computer-assisted verification of a protocol for certified email. In International Static Analysis Symposium (pp. 316-335). Springer, Berlin, Heidelberg.

[92] Kremer, S., Ryan, M., & Smyth, B. (2010, September). Election verifiability in electronic voting protocols. In European Symposium on Research in Computer Security (pp. 389-404). Springer, Berlin, Heidelberg.

[93] Abadi, M., Blanchet, B., & Fournet, C. (2007). Just fast keying in the pi calculus. ACM Transactions on Information and System Security (TISSEC), 10(3), 9-es.

[94] Blanchet, B., Smyth, B., Cheval, V., & Sylvestre, M. (2018). ProVerif 2.00: automatic cryptographic protocol verifier, user manual and tutorial. Version from, 05-16.

[95] Hansen, K., Larsen, T., & Olsen, K. (2010). On the efficiency of fast RSA variants in modern mobile phones. arXiv preprint arXiv:1001.2249.

[96] Bérczes, A., ÁDÁMKÓ, É. C., Folláth, J., & Pethõ, A. (2013). U.S. Patent Application No. 13/673,085.

[97] Drury, B. (2001). Control techniques drives and controls handbook (No. 35). IET.

[98] Modbus.org, „Modbus FAQ," 2005 [Online].http://www.modbus.org/faq.php.

[99] Ross, R. S. (2014). Assessing security and privacy controls in federal information systems and organizations: building effective assessment plans (No. Special Publication (NIST SP)-800-53A Rev 4).

[100] CISA, „Critical Infrastructure Sectors," 2020 [Online].https://www.cisa.gov/critical-infrastructure-sectors.

[101] ISO, I. (1992). Open Systems Interconnection-Basic Reference Model. ISO/TC, 97, 7498-1.

[102] Network, A. L. A., Layer, D. L., & Layer, P., „ The Ethernet.," 1980.

[103] Kordik, J. „Fundamental Guide to Industrial Networking," 2017.

[104] modbus.org, „MODBUS over Serial LIne Specification & Implementation guide," 2002.

[105] modbus.org, „MODBUS application protocol specifications," 2012.

[106] modbus.org, „MODBUS MESSAGING ON TCP/IP IMPLEMENTATION GUIDE V1.0b," 2006.

[107] Buchanan, B. (2010). The Handbook of Data Communications and Networks: Volume 1 (Vol. 2). Springer Science & Business Media.

[108] Kugelstadt,T. The RS-485 Design Guide, Texas Instruments, 2008.

[109] Boyer, S. A. (2009). SCADA: supervisory control and data acquisition. International Society of Automation.

[110] Smith, A. B. (1994). IEEE std c37. 1-1994, IEEE standard definition, specification, and analysis of systems used for supervisory control, data acquisition, and automatic control. IEEE Power Engineering Society, Sponsored by the Substations Committee, Institute of Electrical and Electronics Engineers.

[111] Carke, G., Rynders, D., & Wright, E. (2003). Practical Modern SCADA Protocols. Elsevier.

[112] Raghvendra, N. „ SCADA System – Components, Hardware & Software Architecture, Types." Electricalfundablog, 2015 [Online]. https://electricalfundablog.com/scada-system-components-architecture/#Applications_of_SCADA_System.

[113] Krambeck, D. „ An Introduction to SCADA Systems," Allaboutcircuits, August 2015. [Online]. https://www.allaboutcircuits.com/technical-articles/an-introduction-to-scada-systems/.

[114] IVCM156308, M. T. B. A COMPARATIVE ANALYSIS OF HEALTHCARE SYSTEM IOT AND INDUSTRIAL SCADA IOT FOR CYBERTERRORISM.

[115] Adamko, E., Szemes, P., & Mihoko, N. (2014). Investigation on the heating system of the mechatronics research center building using olap technology. Environmental Engineering & Management Journal (EEMJ), 13, 2733-2742.

[116] „SCADA Systems and Products," DCC, 2019 [Online]. https://digitalcc.com/scada/.

[117] Corley, M., „Monitoring Water and Wastewater Systems with SCADA," Indusoft Web Studio , 26 05 2017. [Online]. https://www.indusoft.com/blog/2017/05/26/monitoring-water-and-wastewater-systems-with-scada/.

[118] „Critical Infrastructure," Panda, 2018 [Online]. https://www.pandasecurity.com/mediacenter/src/uploads/2018/10/1611-WP-CriticalInfrastructure-EN.pdf.

[119] Turk, R. J. (2005). Cyber incidents involving control systems (No. INL/EXT-05-00671). Idaho National Laboratory (INL).

[120] Nash, T. „Backdoors and holes in network perimeters," ICS-Cert, 2005. [Online].http://ics-cert. us-cert. gov/controlsystems.

[121] Hemsley, K., & Fisher, R. (2018, March). A history of cyber incidents and threats involving industrial control systems. In International Conference on Critical Infrastructure Protection (pp. 215-242). Springer, Cham.

[122] Ackerman,R. K., „SCADA Systems Face Diverse Software Attack Threats.," Signal, 2013. [Online]. https://www.afcea.org/content/scada-systems-face-diverse-software-attack-threats.

[123] Reed, T. C. (2005). At the abyss: an insider's history of the Cold War. Presidio Press.

[124] Denning, D. E. (2000). Cyberterrorism: The logic bomb versus the truck bomb. Global Dialogue, 2(4), 29.

[125] Markey,E. J., „Infection of the Davis Besse Nuclear Plant by the "Slammer" Worm Computer Virus - Follow-up Questions," U.S. NRC, 2003. [Online]. https://www.nrc.gov/docs/ML0329/ML032970134.pdf.

[126] Moyer, M. „Expert: A Virus Caused the Blackout of 2003. Will the Next One Be Intentional?," Scientific American, 2011. [Online]. https://blogs.scientificamerican.com/observations/expert-a-virus-caused-the-blackout-of-2003-will-the-next-one-be-intentional/.

[127] Langner, R., „To kill a centrifuge," Langner, 2013. [Online]. https://www.langner.com/to-kill-a-centrifuge/.

[128] Fruhlinger, J., „What is stuxnet who created it anf how does it work," CSO, 2017. [Online]. https://www.csoonline.com/article/3218104/what-is-stuxnet-who-created-it-and-how-does-it-work.html.

[129] Keizer, G. (2011). 'Sloppy'Chinese hackers scored data-theft coup with 'Night Dragon'. Computerworld, February, 11.

[130] „TOP 2019 CYBER ATTACKS ON ICS (INFOGRAPHIC)," Waterfall, 12 2019. [Online]. https://waterfall-security.com/top-2019-attacks-on-ics/.

[131] Byres, E. J., Franz, M., & Miller, D. (2004, December). The use of attack trees in assessing vulnerabilities in SCADA systems. In Proceedings of the international infrastructure survivability workshop (pp. 3-10). Citeseer.

[132] Nardone, R., Rodríguez, R. J., & Marrone, S. (2016, December). Formal security assessment of Modbus protocol. In 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST) (pp. 142-147). IEEE.

[133] Chen, B., Pattanaik, N., Goulart, A., Butler-Purry, K. L., & Kundur, D. (2015, May). Implementing attacks for modbus/TCP protocol in a real-time cyber physical system test bed. In 2015 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR) (pp. 1-6). IEEE.

[134] Huitsing, P., Chandia, R., Papa, M., & Shenoi, S. (2008). Attack taxonomies for the Modbus protocols. International Journal of Critical Infrastructure Protection, 1, 37-44.

[135] Schneier, B. (1999). Attack trees. Dr. Dobb's journal, 24(12), 21-29.

[136] Saini, V., Duan, Q., & Paruchuri, V. (2008). Threat modeling using attack trees. Journal of Computing Sciences in Colleges, 23(4), 124-131.

[137] Urrea, C., Morales, C., & Muñoz, R. (2016). Design and implementation of an error detection and correction method compatible with MODBUS-RTU by means of systematic codes. Measurement, 91, 266-275.

[138] Yüksel, Ö., den Hartog, J., & Etalle, S. (2016, April). Reading between the fields: practical, effective intrusion detection for industrial control systems. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (pp. 2063-2070).

[139] Solomakhin, R., Tsang, P., & Smith, S. (2010, March). High security with low latency in legacy SCADA systems. In International Conference on Critical Infrastructure Protection (pp. 63-79). Springer, Berlin, Heidelberg.

[140] Transceiver, S. E. (2005). SEL-3021 Serial Encrypting Transceiver Security Policy.

[141] Moore, T., & Shenoi, S. (Eds.). (2010). Critical Infrastructure Protection IV: Fourth Annual IFIP WG 11.10 International Conference on Critical Infrastructure Protection, ICCIP 2010, Washington, DC, USA, March 15-17, 2010, Revised Selected Papers (Vol. 342). Springer Science & Business Media.

[142] Ádámkó, É., Jakabóczki, G., & Szemes, P. T. (2018). Proposal of a Secure Modbus RTU communication with Adi Shamir's secret sharing method. International Journal of Electronics and Telecommunications, 64(2), 107-114.

[143] Maurer, U. M., & Schmid, P. E. (1994, November). A calculus for secure channel establishment in open networks. In European Symposium on Research in Computer Security (pp. 173-192). Springer, Berlin, Heidelberg.

[144] Van Tilborg, H. C., & Jajodia, S. (Eds.). (2014). Encyclopedia of cryptography and security. Springer Science & Business Media.

[145] Cornell University, „System Security Course: Secure Channel," 2019. [Online].www.cs.cornell.edu/courses/cs5430/2016sp/l/09-channel/lec.pdf.

APPENDIX A
LIST OF PAPERS RELATED TO THE THESIS POINTS WITH CITATIONS

1. **Csernusné Ádámkó Éva**, Pethő Attila “Helyszín bélyegzés”, hitelesített GPS koordináták In: Lóki, J (szerk.) Az elmélet és gyakorlat találkozása a térinformatikában II. : II. Térinformatikai Konferencia és Szakkiállítás Debrecen Debrecen, Magyarország : DE TTK Földrajzi Tanszékcsoport, (**2011**) pp. 381-387. , 7 p.

2. **Ádámkó**, **E**., & Pethő, A. (**2013**). Location-stamp for GPS coordinates. Acta Universitatis Sapientiae, Informatica, 5(1), 63-76.

3. Bérczes, A., **ÁDÁMKÓ, É. C.**, Folláth, J., & Pethõ, A. (**2013**). U.S. Patent Application No. 13/673,085.

    - Freeze-Skret, J. (2016). U.S. Patent No. 9,432,390. Washington, DC: U.S. Patent and Trademark Office.
    - Kostiainen, K. (2017). U.S. Patent No. 9,787,667. Washington, DC: U.S. Patent and Trademark Office.
    - Cohen, R. H. (2017). U.S. Patent No. 9,800,415. Washington, DC: U.S. Patent and Trademark Office.
    - Jones, R. K., Steger, C., Brachet, N., Alizadeh-Shabdiz, F., Broadstone, A., & Morrin, J. (2017). U.S. Patent No. 9,817,101. Washington, DC: U.S. Patent and Trademark Office.
    - Salmela, P., & Fornehed, J. (2018). U.S. Patent Application No. 15/551,683.

4. **Ádámkó Éva**, Jakabóczki Gábor Security analysis of Modbus RTU pp. 5-11. In: Kocsis, Imre (szerk.) Proceedings of the Conference on Problem-based Learning in Engineering Education Debrecen, Magyarország: University of Debrecen Faculty of Engineering, (**2015**) p. 98

5. Jakaboczki, G., & **Adamko, E.** (**2015**). Vulnerabilities Of Modbus RTU Protocol–A Case Study. Annals Of The Oradea University, Fascicle Of Management And Technological Engineering, (1).

    - Muñoz, N., & Davensor, C. (2016). Explotando vulnerabilidades en el protocolo MODBUS TCP/IP.

- Dogaru, D. I., & Dumitrache, I. (2017, May). Robustness of Power Systems in the Context of Cyber Attacks. In 2017 21st International Conference on Control Systems and Computer Science (CSCS) (pp. 506-512). IEEE.
- Tranca, D. C., Banu, C. I., & Rosner, D. (2018). EGIFM--Extendable Gateway and Industrial Firewall for ModBus. eLearning & Software for Education, 4.
- Urdaneta Velasquez, M. (2018). Attaques informatiques sur le réseau de contrôle du trafic routier (Doctoral dissertation, École Polytechnique de Montréal).
- Velasquez, M. U. (2018). Attaques Informatiques sur le Réseau de Contrôle du Trafic Routier (Doctoral dissertation, Ecole Polytechnique, Montreal (Canada)).

6. Jakabóczki G, Szemes P T, **Ádámkó É** A MODBUS RTU protokoll biztonságtechnikai vizsgálata, új kriptográfiai megoldások tesztelése = Security evaluation of MODBUS RTU protocol, testing new cryptographic methods INTERNATIONAL JOURNAL OF ENGINEERING AND MANAGEMENT SCIENCES / MŰSZAKI ÉS MENEDZSMENT TUDOMÁNYI KÖZLEMÉNYEK 1: 2 pp. 35-42. , 8 p. (**2016**)
7. **Ádámkó, É.** (**2017**). Security analysis of a „Location-stamping" protocol for GPS coordinates. Műszaki és Menedzsment Tudományi Közlemények, 2(2), 1-12.
8. **Ádámkó, É.**, Jakabóczki, G., & Szemes, P. T. (**2018**). Proposal of a Secure Modbus RTU communication with Adi Shamir's secret sharing method. International Journal of Electronics and Telecommunications, 64(2), 107-114.

- Chromik, J. J., Remke, A., & Haverkort, B. R. (2018). An integrated testbed for locally monitoring SCADA systems in smart grids. Energy Informatics, 1(1), 56.
- Volkova, A., Niedermeier, M., Basmadjian, R., & de Meer, H. (2018). Security challenges in control network protocols: A survey. IEEE Communications Surveys & Tutorials, 21(1), 619-639.
- Tidrea, A., Korodi, A., & Silea, I. (2019). Cryptographic Considerations for Automation and SCADA Systems Using Trusted Platform Modules. Sensors, 19(19), 4191

- Gamess, E., Smith, B., & III, G. F. PERFORMANCE EVALUATION OF MODBUS TCP IN NORMAL OPERATION AND UNDER ADistributed DENIAL OF SERVICE ATTACK. International Journal of Computer Networks and Communications 12(2):1-21 (2020)

APPENDIX B
ACKNOWLEDGEMENTS

Foremost, I would like to thank God; without His grace and blessings, this research would not have been possible to complete.

I would like to express my appreciation to my supervisor Prof. Attila Pethő, for his patience and great ideas during the last decade.

Besides the above, I would like to express my special gratitude to my dearest friend and colleague Dr. Síki Gusztáv Áron for his tireless and tenacious help during the last two years—also, the many early mornings and late nights when he was always ready to answer my questions. Furthermore, the support that he provided me with his wise advice and reassuring presence.

Last but not least, I would like to thank my husband and my two children for supporting me throughout the years of my PhD studies and their endless patience and kindness.