İstanbul Commerce University Journal of Science, 23(45), Spring 2024, 114-132.

İSTANBUL TİCARET ÜNİVERSİTESİ
FEN BİLİMLERİ DERGİSİ

İstanbul Commerce University Journal of Science

http://dergipark.org.tr/ticaretfbd

Research Article / Araştırma Makalesi

USING ELLIPTIC CURVE CRYPTOGRAPHY FOR AUTHENTICATION AND KEY EXCHANGE IN CONSTRAINED INTERNET OF THINGS NETWORKS*

KISITLI NESNELERİN İNTERNETİ AĞLARINDA ELİPTİK EĞRİ ŞİFRELEMENİN DOĞRULAMA VE ANAHTAR KARŞILAŞTIRMA AŞAMASINDA KULLANILMASI

İbrahim KARATAŞ[1] **Selim BAYRAKLI[2]**

https://doi.org/10.55071/ticaretfbd.1439890

Corresponding Author / Sorumlu Yazar
hbayrakli@hho.msu.edu.tr

Received / Geliş Tarihi
19.02.2024

Accepted / Kabul Tarihi
07.04.2024

Abstract

It is anticipated that billions of objects will be interconnected with the rise of the Internet of Things, leading to the evolution of the Internet for the upcoming generation. Various applications have been created in different sectors such as health, logistics, industry, and military in recent years. The techniques created for IoT are still in a nascent stage and encounter numerous hurdles. The primary concern is the security issue. These devices are a significant target due to the numerous conveniences offered by the Internet of Things. These gadgets will maintain continuous communication with one other (M2M) and with people (M2H). It is crucial to ensure the safe transmission of key information about people and the environment throughout this communication. Today's security approaches cannot be integrated into Internet of Things networks because of constraints such as limited RAM, ROM ratio, low bandwidth, poor computing power, and low energy supply. The DTLS protocol, created by IETF, utilizes symmetric encryption and may not be suitable for Class-0 and Class-1 devices that require asymmetric encryption. This study examines the security measures in place and the data is securely exposed to the internet using Elliptic Curve Cryptography, then compared with other studies.

Keywords: Cryptography, encryption, elliptic curve, internet of things, security.

Öz

Nesnelerin İnternetinin ortaya çıkmasıyla birlikte milyarlarca nesnenin birbirine bağlanacağı ve gelecek nesil için İnternetin evrimine yol açacağı öngörülmektedir. Son yıllarda sağlık, lojistik, endüstri ve askeri gibi farklı sektörlerde çeşitli uygulamalar oluşturulmuştur. IoT için oluşturulan teknikler henüz başlangıç aşamasındadır ve çok sayıda engelle karşılaşmaktadır. Bunların başında güvenlik sorunu gelmektedir. Nesnelerin İnterneti tarafından sunulan sayısız kolaylık nedeniyle bu cihazlar önemli bir hedeftir. Bu aygıtlar birbirleriyle (M2M) ve insanlarla (M2H) sürekli iletişim halinde olacaktır. Bu iletişim boyunca insanlar ve çevre hakkında önemli bilgilerin güvenli bir şekilde iletilmesini sağlamak çok önemlidir. Günümüzün güvenlik yaklaşımları, sınırlı RAM, ROM oranı, düşük bant genişliği, zayıf hesaplama gücü ve düşük enerji kaynağı gibi kısıtlamalar nedeniyle Nesnelerin İnterneti ağlarına entegre edilememektedir. IETF tarafından oluşturulan DTLS protokolü simetrik şifreleme kullanmaktadır ve asimetrik şifreleme gerektiren Sınıf-0 ve Sınıf-1 cihazlar için uygun olmayabilir. Bu çalışma, mevcut güvenlik önlemlerini incelemekte ve verilerin Eliptik Eğri Kriptografisi kullanılarak güvenli bir şekilde internete maruz kalmasını sağlamakta, ardından diğer çalışmalarla karşılaştırmaktadır.

Anahtar Kelimeler: Eliptik eğri, güvenlik, nesnelerin interneti, kriptografi, şifreleme.

*This publication is derived from the Master's thesis of İbrahim KARATAŞ, Maltepe University, Institute of Graduate Studies in Sciences, Computer Engineering Program.

[1]Ozden Cengiz Anatolian High School, İstanbul, Türkiye.
ikaratas2515@gmail.com, Orcid.org/0000-0002-5558-3691.

[2]Turkish National Defence University, Air Force Academy, Department of Computer Engineering, Istanbul, Türkiye.
hbayrakli@hho.msu.edu.tr, Orcid.org/0000-0003-3115-6721.

1. INTRODUCTION

With advancing technology, wireless sensor networks have emerged, consisting of sensor nodes that communicate with each other over short distances via wireless connectivity and perform many customized tasks such as collecting, processing and transmitting data. These structures are equipped with limited resources (amount of Random Access Memory-RAM, Read Only Memory-ROM, battery capacity and processing power) and usually have one or more base stations; they are used in a wide range of applications from healthcare to construction, chemistry to environmental monitoring, surveillance of battlefields, factory automation to remote control of homes.

The concept of the Internet of Things (IoT) has emerged with the idea that everything possible can be connected to the internet and therefore to each other. This network, called the internet of the next generation, is formed by the integration of many technologies, including sensor structures, communication, networking, and intelligent information processing technologies (Aloul et al., 2015). In this network structure, which will connect billions of devices, information will be collected through sensors and transferred to the internet environment, where it will be processed and used in various decision-making structures. These devices, which will be used so widely, will have to cope with many security threats found on the traditional internet due to limitations such as processing power, battery life, bandwidth, etc. Therefore, the most important obstacle to the development of the Internet of Things is the security problem. Considering that a lot of sensitive information will be carried over these networks and devices, it will be better understood how sensitive security is. This need for security increases, especially in healthcare and military applications. For example, the insulin pump used in the health sector is one of the best examples of this situation. The insulin pump regularly measures the patient's sugar level and injects the required amount of insulin into the patient when necessary, ensuring that the blood sugar level is at the desired level. In addition, these devices are connected to the internet and send the patient's sugar measurements as a report. It is clear that if such devices are not secured, they can have major consequences.

There are some conditions that are necessary to ensure security in communication between two nodes. These are data confidentiality, data integrity, source authentication, data timeliness, key authentication, and service continuity. The methods used on the traditional internet to ensure these conditions are not possible due to the limitations of wireless sensor devices. Therefore, either separate protocols and new encryption techniques need to be developed for these devices, or existing technologies need to be mitigated. Security services such as authentication and key management are critical to the communication process in wireless sensor networks. In today's traditional networks, such as the Internet, public key cryptography (PKC) allows the use of many security services and protocols (e.g., Secure Socket Layer-SSL, IPsec, and Transport Layer Security-TLS) to ensure secure communication. For example, many security services and protocols (e.g. SSL, IPsec, TLS) often utilize public key cryptography for the distribution of symmetric keys and multi-user authentication messages. However, these public key encryption-based security methods have not yet been fully implemented on resource-constrained sensor devices for the reasons mentioned.

In 1985, Neal Koblitz and Victor Miller first proposed elliptic curve cryptology. Elliptic Curve Cryptology (ECC) is based on the difficulty of the discrete logarithm problem. It provides the same level of security as RSA, but with a much lower key length. For this reason, significant work has been done on the use of ECC in resource-constrained devices.

The aim of this study is to ensure that the limited devices in the Internet of Things (IoT) communicate with each other (Machine-to-Machine - M2M) and with humans (Machine-to-

Human - M2H) and exchange data in a secure manner. In doing so, constraints such as processing power, energy consumption, RAM, and ROM capacity are taken into account, and a method is proposed. It has been observed in studies that, especially in the handshake phase (authentication and key management) of Data Block Transport Layer Security (DTLS), limited devices are overloaded and problems arise in ensuring security. In this study, elliptic curve encryption is used to enable nodes to authenticate each other and exchange keys, and it is shown that asymmetric encryption can be used in resource-constrained nodes. In addition, the developed elliptic curve encryption libraries, RSA, and symmetric encryption methods are compared in terms of the amount of memory (RAM and ROM) they use. The study was implemented and evaluated in a Cooja simulation environment on the Contiki operating system.

The paper is organized as follows: Section 2 discusses the importance of security in the Internet of Things and the general aspects of security. In Section 3, some of the similar studies in the literature on the subject are mentioned. Section 4 summarizes the topic of elliptic curve encryption. Section 5 summarizes the structure and security protocols of the Internet of Things. Section 6 describes the use of the simulation environment in the authentication phase of elliptic curve encryption. And finally, Section 7 provides information about the results obtained.

2. INTERNET OF THINGS AND SECURITY

Devices in the Internet of Things must be connected to each other and to the Internet to perform tasks like sensing the environment, communicating, collecting information, and processing the obtained data. The Internet of Things has the ability to acquire, transmit, and process information from end nodes (Li & Xu, 2017). Having these important features raises the security problem to the highest level, according to the sensitivity of the information carried through sensor networks. This makes it mandatory to ensure the security of these sensor nodes, which are frequently used, especially in the military and health fields. Overcoming the security problem, which is seen as the biggest obstacle in the development of the Internet of Things, is essential for the widespread adoption of IoT.

The security vulnerability arising from the inherent system limitations of sensor nodes cannot be overcome with the security mechanisms available on the traditional Internet. These devices can expose important information (personal, military, health, etc.) carried through them in the event of an attack due to the lack of security mechanisms. This shows that these devices, which lack encryption and authentication, can easily fall into the hands of others. Figure 1 shows the structure and general working logic of the Internet of Things.

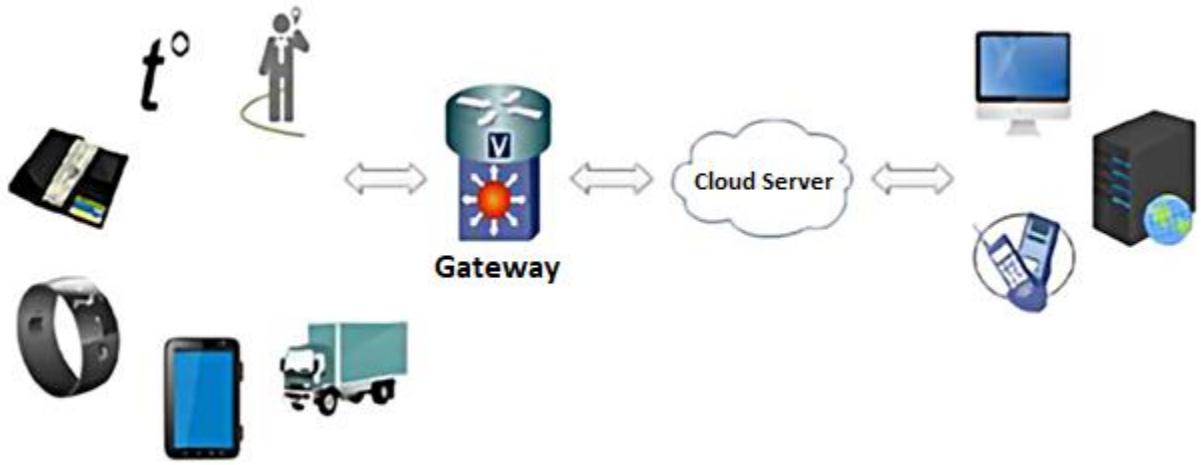

Figure 1. Structure of the Internet of Things Ecosystem

However, on the Internet of Things, many devices lack the resources to ensure secure data communication. These devices are currently in use and continue to exchange data without security measures. In their 2014 study, Bornmann et al. classified these devices in Table 1 according to the amount of RAM and ROM they have (Bormann et al. 2014).

Table 1. Classification of Energy-Constrained Devices

Class	RAM	ROM
Class-0	<<10 KB	<<100 KB
Class-1	~ 10 KB	~100 KB
Class-2	~ 50 KB	~250 KB

The classification of these devices determines which security mechanism to use on which class of device. Class-2 devices can easily integrate currently used security methods. It has sufficient resources for CoAP at the application layer, and DTLS is used for transmission layer security. However, Class-1 and Class-0 devices do not have the resources to fully support these protocols. Especially the handshake phase (authentication and handshake management) in the DTLS protocol at the transmission layer consumes a significant amount of resources, and therefore Class-0 and Class-1 devices cannot use this security protocol. An intervention at this stage to reduce the processing intensity will secure communication between these resource-constrained devices.

2.1. Security

Security ensures that unauthorized persons cannot intercept data while it is being transmitted from one end to the other, or if intercepted, the data remains incomprehensible to third parties. In order to say that security is ensured, the issue needs to be addressed in all its dimensions, as shown in Figure 2.

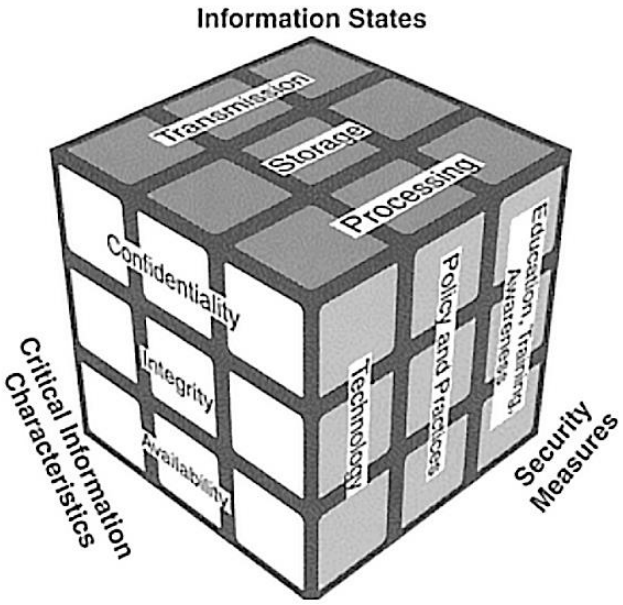

Figure 2. McCumber Cube (McCumber, 2004)

Confidentiality is the prevention of unauthorised access to information. Both the storage phase and the transmission phase should ensure the confidentiality of information. The McCumber model states that information should be encrypted both during storage and transmission (McCumber, 2004).

Integrity means that the content of the information cannot be changed during storage or transmission, and its integrity is preserved. A hash function ensures the integrity of the information by creating a short fingerprint (summary) of the data. This function creates a short fingerprint (summary) of the data, and when the data is changed, it is understood that it has been changed since this fingerprint will not be valid. Let h be a hash function and x be data. The summary of the data is y = h(x). Here, y is stored in a secure location, but x is not. If the message x is modified, then the hash function of the message xl is taken: y = h(xi). In this case, y ^ y indicates that the message content has been modified. In order to check that the data has not been altered during the storage phase, fingerprints are calculated from time to time, and it is checked that it has not been altered. Examples of hash functions are MD5, SHA-1, and HAVAL (Stinson, 2005).

Authorized persons can access data whenever they need it. Information is in a state of constant change and therefore must be constantly accessible. Even in the event of system damage, it must be repaired and made operational as quickly as possible. Extra security equipment and software, such as firewalls and proxy servers, can protect against downtime and inaccessible data due to malicious actions such as denial of service (DoS) attacks and network intrusions.

In addition to these three important elements, requirements such as data timeliness, service integrity, heterogeneity, and key management need to be met. Messages transmitted in the network must be new and not repeated to ensure data timeliness. To ensure this, each message on the network is timestamped. Service integrity is the ability to securely collect data from nodes without corruption. Heterogeneity is the ability of devices with different features developed by manufacturers on the Internet of Things to communicate with each other. Key management: In the Internet of Things, devices need to exchange some parameters between each other to ensure the security of the data. Simplified key management and minimum energy-consuming key distribution methods should be used for these security mechanisms.

3. LITERATURE REVIEW

Liu & Ning (2008) presented TinyECC, an open source software for elliptic curve-based public key cryptography in wireless sensor networks that can be tuned according to different applications as the resources of the resource-constrained device allow. TinyECC can be tuned according to requirements such as memory usage, runtime, and resource consumption. Due to resource consumption, the study was carried out on the Wismote node. This work integrates three standard elliptic curve algorithms: the Elliptic Curve Diffie Hellman Key Agreement Algorithm (ECDH), the Elliptic Curve Digital Signature Algorithm (ECDSA), and the Elliptic Curve Integrated Encryption Scheme (ECIES).

Szczechowiak et al. (2008) showed in NanoECC that elliptic curve cryptology can be used for resource-constrained devices. However, according to the results obtained, it is still not completely satisfactory. However, the MICA2 (8-bit/7.3828 MHz ATmega128L) and Tmote Sky (16-bit/8.192 MHz MSP-430) nodes yielded the most effective results.

In the study by Zhao & Ge (2013), security problems and solutions in the three-tier system architecture of the Internet of Things are mentioned. Among these security measures, key management and algorithms in the sensing layer, security routing protocol authentication, and access control are also emphasized. Researchers have attempted to overcome the security problems in devices with limited resources using Internet of Things technology by employing symmetric and asymmetric encryption techniques. It is revealed that the security problem is not a single-layer problem but a problem of all layers.

Santos et al. (2015) stated that the security problem poses the biggest obstacle in the development of IoT and that limited processing and memory capacity do not support standard security mechanisms. In the developed architecture, the use of Data Block Transport Layer Security (DTLS) is enabled by using a third-party device called the Internet of Things Security Support Provider (IoTSSP) for mutual authentication and communication on constrained devices.

In Lithe (Low Load CoAP Security in loT) by Raza et al. (2013), a study on the application of DTLS and CoAP to the Internet of Things was carried out. In addition, they proposed a DTLS header compression scheme that significantly reduces energy consumption. This header compression mechanism did not affect end-to-end security. Simultaneously, the DTLS operation significantly reduced the number of exchanged bytes. The Contiki operating system was used for the DTLS-based implementation. The results show significant gains in packet size, energy consumption, processing time, and network response time when the compressed DTLS mechanism is activated.

There is an increasing need for utilizing suitable cryptographic methods in embedded applications inside an IoT setting with several connected intelligent devices, as indicated by the research conducted by Aazam et al. (2016).

IoT devices in the study by Nakagawa & Shimojo (2017) typically perform both IoT and device activities and face limitations in resources, which hinders the integration of security controls on these devices. The authors suggested an agent-based security approach to prevent unauthorized access to physical devices by malevolent users, which involved separating the IoT function from the device function. This project aims to separate IoT functionalities from devices and integrate them into a cloud environment. The authors of the article suggested an architectural framework for IoT agents where virtual replicas of IoT devices operate in a cloud setting.

In the study titled Security for Internet of Things Technology published by Görmüş et al. (2017), the current protocols considered for the Internet of Things are introduced, and the security vulnerabilities that may occur are examined for each layer. It is stated that protocols such as 6LoWPAN, 6TISCH, and CoAP should be designed with an understanding that security is at the forefront.

4. CRYPTOLOGIC METHODS

Cryptology is a Greek word meaning secret writing. The basis of cryptology is that two people communicating through an insecure channel, as in Figure 3, cannot understand what is happening even if a third party intercepts the message.

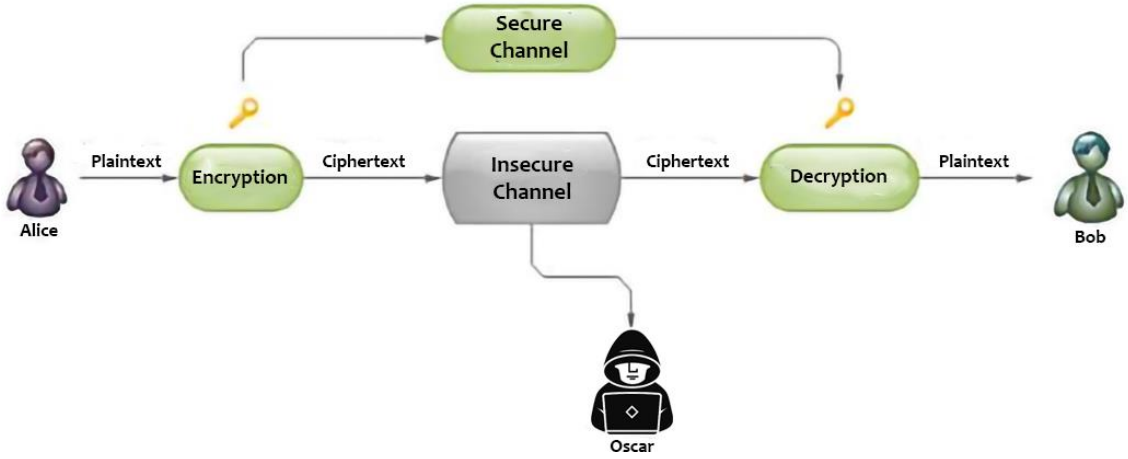

Figure 3. Encryption over Insecure Channel

A cryptosystem is a group of 5 definitions (P,C,K,E,D) satisfying the following conditions;

1. P; finite set of possible plaintexts,
2. C; finite set of possible ciphertexts,
3. K is the finite set of all possible keys,
4. $\forall k \in K$, according to the encryption rule there are $e_k \in E$ and $d_k \in D$. $e_k : P \rightarrow C$ and $d_k: C \rightarrow P$ are functions. Thus $d_k(e_k(x)) = x$ for $\forall x \in P$.

Here, the plaintext x is encrypted using the ek function and decrypted using the dk function. By the way, it is obvious that the ek function is one-to-one. Otherwise, there is no inverse of the function, and the decryption cannot be performed (Stinson, 2005).

4.1. Elliptic Curve Cryptography

Elliptic curve cryptography is an asymmetric cryptosystem based on the algebraic topology of elliptic curves over finite fields. The reliability of ECC is based on the difficulty of solving the elliptic curve discrete logarithm problem. It can be used in many areas, such as key agreements and digital signatures. An elliptic curve is a curve defined over the set of real numbers that satisfies the general equation $y^2 = x^3 + ax + b$ for the real numbers x and y. For this general equation, each value of a and b gives a different curve (Trappe & Washington, 2005).

The numbers a and b in this equation are real numbers, and the equation $x^3 + ax + b$ must have $4a^3 + 27b^2 \neq 0$ to have no multiple roots. Compared to a public-key cryptosystem, elliptic curve cryptography uses a shorter key length for the same level of security. Elliptic curve cryptography can provide the same level of security as RSA with a 1536-bit key size, but with a much shorter

160-bit key. This is very useful for embedded systems and smartphones. Here is how the elliptic curve algorithm works:

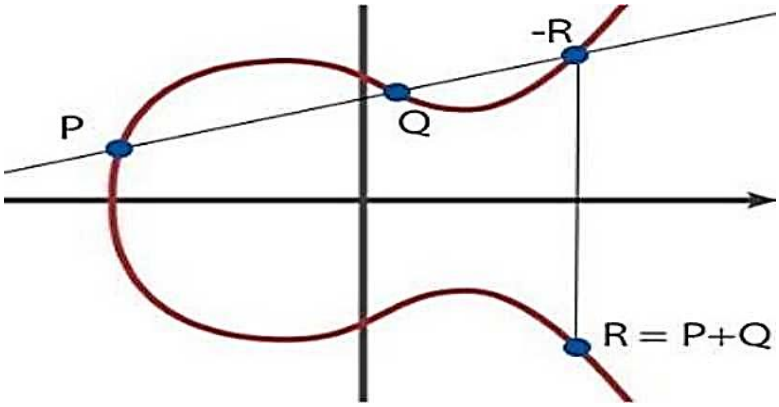

Figure 4. Summation of Two Points on an Elliptic Curve (Orhon, 2015)

p is a prime exponential number, and Fp is a finite field containing p elements. The result of any operation defined on this set is also an element of this field (mod p). The elliptic curve discrete logarithm problem is to find the unique solution k, where $0 \leq k \leq p-1$ to the equation Q=kP. This is given an elliptic curve E in the domain Fp and two points P and Q on it. Let Q = kP when P, Q ∈ EF (a,b) and k < p. While it is relatively easy to calculate Q given k and P, it is really hard to calculate k given Q and P (Trappe & Washington, 2005; Bozkurt, 2005).

Here P and Q are two points on the curve, and a line drawn from these points must intersect the curve at a third point, -R. The symmetry of the point -R with respect to the x-axis is also a point R on the curve. In this case, we obtain $P+Q = R$, as shown in Figure 2.9. When this process is repeated k times, another point on the curve is obtained. Knowing the starting point P and the number of times this process has been executed makes it easy to obtain the point Q, which is why this is considered a nice trapdoor function. But once P and Q are known, k is almost impossible to get.

4.1.1 Elliptic Curve Diffie Hellman key exchange algorithm

The Elliptic Curve Diffie-Hellman Key Exchange Algorithm is a key exchange protocol that allows two parties to agree on a key using elliptic curve encryption over an insecure channel. It works as follows:

i. Key Production Phase

- The sender and receiver agree on the parameters p,a,b,G,n,h.
- The private keys d_s and d_r are chosen randomly in the range [1,n-1].
- The public keys are $e_S = d_S G$ and $e_R = d_R G$.

ii. Calculation of the Private Key

- The sender calculates $K = (x_K, y_k) = d_S e_R$.
- The receiver calculates $L=(x_L, y_L) = d_R e_S$.
- $D_S e_R = d_S d_R G = d_R d_S G = d_R e_S$
- Thus K=L and $x_K = x_L$.
- The private key is x_k (Franco, 2024).

The elliptic curve encryption method significantly reduces the load caused by asymmetric encryption in Diffie-Hellman key exchange. Especially in the use of symmetric and asymmetric encryption methods together, which we call hybrid systems, the key exchange problem arising from symmetric encryption will be overcome with a very small load with the elliptic curve Diffie-Hellman key exchange protocol, and problems such as computation, memory, and battery, especially in resource-limited devices, will be overcome. This can provide a solution to the problem of secure communication on the Internet of Things in this respect.

5. THE STRUCTURE OF THE INTERNET OF THINGS

IoT will become a part of people's daily lives as the communication and networking capabilities of smart devices and physical objects expand. These devices are devices that work in cooperation with each other, with the ability to collect and process information for the benefit of people and make decisions when appropriate. As the IoT network structure is shown in Figure 5, sensor devices in many different areas are connected to the internet through different gateways in a wired or wireless manner.

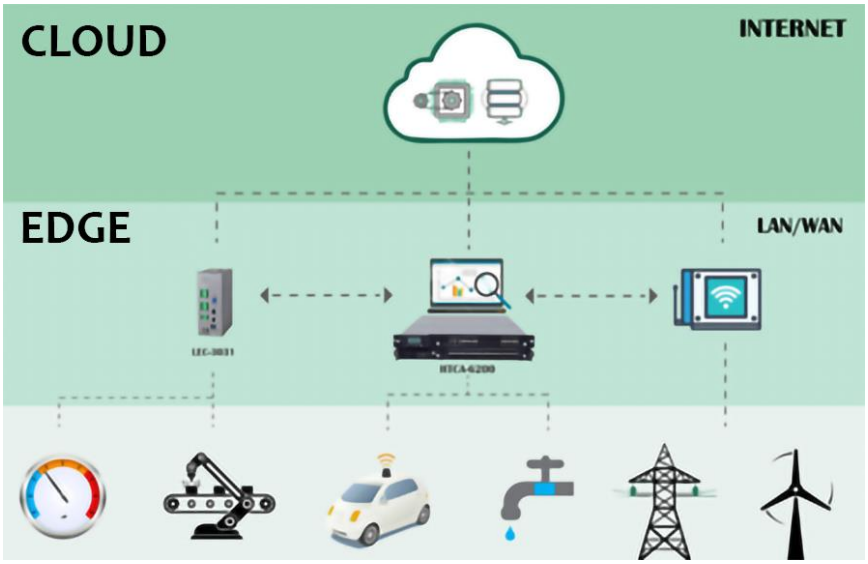

Figure 5. Internet of Things Network Structure (Huawei, 2023)

Using IoT technology means opening up all our personal privacy to the internet, allowing unprecedented access to and information collection about us. This shows how important security is. Security is a multi-dimensional problem. It covers many aspects, from the physical protection of the device to the encryption of the information inside the device, harmonisation of devices belonging to different companies, and ensuring that the system is always operational. The part we are interested in within the scope of this thesis is the encrypted delivery of data to the other party during transmission.

The following conditions should be taken into consideration when designing a safe structure:

1- Technical factors: detection technique, communication method, network technology
2- Security protection; information confidentiality, transmission security, privacy protection (Li & Xu, 2017).

5.1. Constrained Application Protocol (CoAP)

Constrained Application Protocol (CoAP) is an application layer protocol developed by the Internet Engineering Task Force (IETF) for machine-to-machine (M2M) communication on resource-constrained devices. It is a variant of the widely used synchronous internet protocol HTTP. Similar to HTTP, it offers a REST interface but is more system-constrained. In the Internet of Things, nodes are generally 8-bit microcontrollers with a limited amount of RAM and ROM and a high packet loss rate in constrained networks such as 6LoWPAN (low-power wireless personal area networks over IPv6). CoAP has been developed to meet specific requirements such as low power, simplicity, and multiple communications in resource-constrained environments while easily interfacing with HTTP for integration with the web.

Constrained networks such as 6LoWPAN support fragmenting IPv6 packets into small link layer perimeters, but this causes serious problems in packet forwarding. One of the main reasons for designing CoAP is to reduce the need for fragmentation by keeping the message size small. The main features of CoAP are:

1- The Web protocol fully meets the M2M requirement in constrained environments.
2- Supports one-way and multi-way communication.
3- Communicates with asynchronous message exchange.
4- Reduced head load and complexity.
5- Provides simple proxy and caching capacity.
6- Provides reliable connection structure thanks to Datagram Transport Layer Security (DTLS) (Shelby et al., 2014).

CoAP's interactive model is similar to HTTP's client/server structure. However, the M2M interaction structure in CoAP allows devices to act as both servers and clients. CoAP reduces implementation complexity (code size) and bandwidth because it has been developed for constrained networks. This reduction in resource utilization helps to increase reliability (by reducing fragmentation at the link layer) and reduce latency in lossy networks. (Chavan & Nighot, 2014).

CoAP uses a REST service structure similar to HTTP. Because data size and application speed are important, using this structure provides a significant advantage. It must also optimize the data size and ensure reliable communication to overcome the disadvantage of limited devices. On the other hand, CoAP supports URI and REST methods such as GET, POST, PUT, and DELETE. To overcome the weakness of the insecure nature of the UDP (User Datagram Protocol) protocol, CoAP has defined a retransmission mechanism and a resource discovery mechanism. (Chen, 2014).

DTLS encrypts CoAP messages at the transport layer. It is the UDP-based version of TLS used on the traditional Internet. Securing DTLS means securing CoAP as well. Four types of security mechanisms are defined for CoAP. These are NoSec, PreSharedKey, RawPublicKey, and Certificate modes. (Shelby ve ark., 2014).

5.2. Datagram Transport Layer Security (DTLS)

The Transport Layer Security (TLS) protocol secures network traffic, ensuring data integrity and confidentiality between the two communicating ends. The TLS protocol consists of two layers: the TLS Registration Protocol and the TLS Handshake. However, TLS must operate over a reliable transmission channel such as TCP. The DTLS protocol ensures communication security for applications that use UDP as the transmission layer.

Security protocols in Internet of Things applications generally target small, low-power, remotely controllable sensors and components. The DTLS protocol is a transmission layer protocol based on TLS that provides equal security services such as confidentiality, authentication, and integrity protection. Since TLS works over TCP, it does not have to deal with problems such as packet loss and packet ordering. In DTLS, the handshake mechanism has to deal with packet loss, reordering, and retransmission. In DTLS, the initial authentication of nodes, key agreement, and data protection take place over a secure channel.

Datagram transmission does not guarantee the sequential transmission or reliability of data. The DTLS protocol preserves these properties for application data. For applications such as media over the Internet, Internet telephony, or games, UDP is used for communication due to the delay sensitivity of the data transmitted (Rescorla et al., 2022).

The DTLS protocol is a protocol that manages packet loss due to UDP, provides packet reordering at the receiving end, and operates on lower frames.

5.3. IPv6 in Low Power Wireless Personal Area Networks (6LoWPAN)

The IETF created the 6LowPAN protocol as an adaptation layer for devices with the IEEE 802.15.4 physical layer to be included in the Internet (Figure 6). It works as an adaptation layer between the IEEE 802.15.4 layer and the IPv6 layer. The maximum transfer size (MTU) that the IEEE physical layer can send is 127 bytes, and the minimum transfer size (MTU) that IPv6 packets can send is 1280 bytes, making the use of an adaptation layer mandatory. For this reason, the 6LoWPAN layer was developed to compress IP packets into small pieces and transmit them by compressing them. The features of the 6LoWPAN protocol are as follows:

1- Minimal code and memory usage required by the Internet of Things,
2- Long battery life,
3- Efficient IP and TCP/UDP header compression,
4- Supports 16 and 64-bit 802.15.4 addressing,
5- Automatic network configuration with neighbor discovery method,
6- Single, multiple communication and broadcast support,
7- Low power and cost requirement,
8- Header compression, fragmentation and concatenation of IPv6 packets (Li & Xu, 2017).

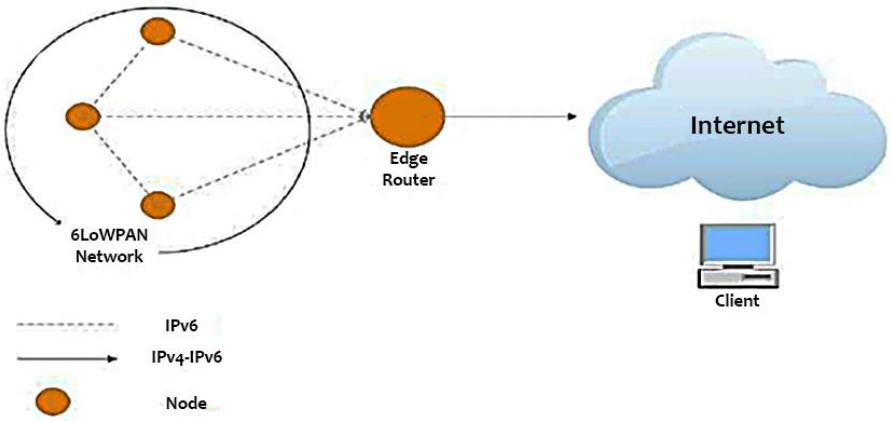

Figure 6. Structure of 6LoWPAN Networks

6. AUTHENTICATION USING ELLIPTIC CURVE IN A SIMULATION ENVIRONMENT

The Cooja simulator on the Contiki operating system installed on the virtual machine is used to realize the use of the elliptic curve encryption algorithm in the Internet of Things environment. The low cost of the Contiki operating system for internet communication is an important feature for resource-constrained devices in the Internet of Things. It has IPv6 and IPv4 support. It also supports 6LoWPAN, RPL, and CoAP protocols developed for the Internet of Things. Contiki applications are written in the standard C language. The Cooja simulator allows for simulating the code before writing it to the hardware (Karataş & Bayraklı, 2020).

6.1. Introduction of the System

In this paper, authentication and key exchange based on elliptic curve encryption are implemented to solve the security problem on the Internet of Things. Asynchronous UDP was chosen as the transmission protocol because the TCP protocol imposes a heavy load on the system, which is a major problem for resource-constrained devices. The DTLS protocol was used to ensure secure communication. For DTLS, the open-source TinyDTLS library was preferred (Bergmann, 2017). The elliptic curve digital signature algorithm and Diffie-Hellman key exchange algorithm are activated during the handshake phase to reduce the overhead caused by asymmetric encryption. This study aims to demonstrate the use of elliptic curve encryption for resource-constrained devices in the Internet of Things and the ability of nodes to communicate end-to-end. Figure 7 illustrates the general structure of the study.

6.2. Node Selection

Since elliptic curve encryption and the DTLS protocol consume a large amount of RAM and ROM and impose a heavy load on the processor, it is important to choose the node to be used. Considering the code size, the Skymote node and the Z1 node have low RAM and ROM amounts, so we used the Wismote node in our study and simulated it with the TinyDTLS library using the Cooja simulation program. Table 2 shows the memory amounts of the nodes.

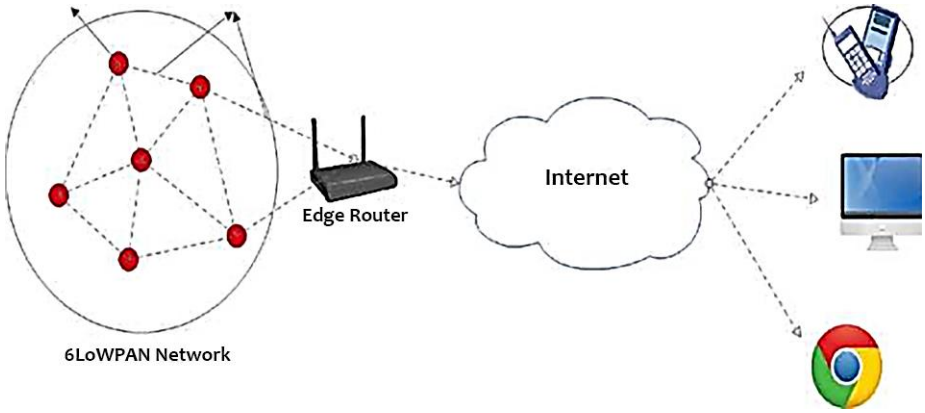

Figure 7. General Structure of the Study

Table 2. Memory Values of Sensor Nodes

	Skymote	Z1mote	Wismote
Ram	10KB	8KB	16KB
Rom	48KB	92KB	128KB
Microcontroller	MSP430	MSP430	MSP430

6.3. Client-Server Communication with DTLS Elliptic Curve Encryption

Figure 8 shows the design of the handshake phase and key exchange simulation using the TinyDTLS library with the Wismote node in the Cooja simulator. The communication phase of the nodes starts with the Hello Client message of the client, which is the first flow. This message contains session ID, header compression method, and password blocks. In response to this message, the server sends a Hello Confirmation Request message in the second flow. A session information message called a cookie is included in this message to provide security against DoS attacks. In the third flow, the client sends the first message again by adding the cookie file, and communication starts. In the 4th flow, the server requests its own information with the Hello Server message, the information required to verify its identity with the Certificate message, its own public key information with the Server Key Exchange message, and the information required to verify the identity of the other party with the Certificate Request message. In the 5th flow message, the client transmits to the server the information needed to verify its identity with the Certificate message, its own public key with the Client Key Exchange message, and the information needed to verify the identity of the other party with the Certificate Confirmation message. The end message completes the authentication and key exchange process.

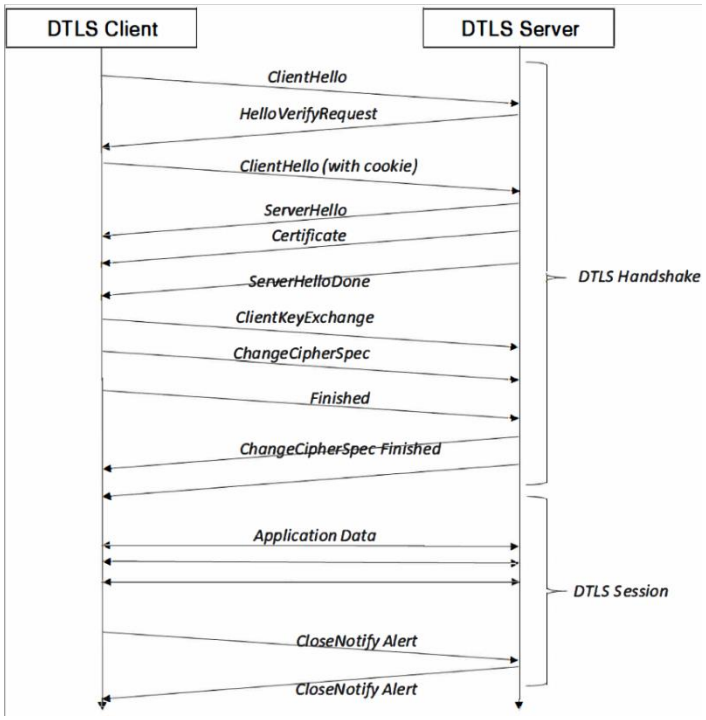

Figure 8. DTLS Handshake Phase Architecture (Microsoft, 2021)

Sensor devices (nodes) in the Internet of Things environment are severely resource-constrained. Therefore, the resource constraints of sensor devices in the Internet of Things environment make it

crucial to consider this aspect in studies. Especially asymmetric encryption algorithms impose a serious burden on the system and consume a significant amount of RAM, ROM, and processor power, and the devices cannot fulfill the purpose for which they were produced. For this reason, the amount of RAM and ROM usage and the amount of energy consumed by the sensor device were measured in this study.

The memory occupied by the compiled code directly affects the type of node we will use. Table 3 shows the amount of memory occupied by the TinyDTLS code used in the study.

Table 3. RAM and ROM Requirements of the System

Text	Data	Bss	Dec	File
83885 byte	422B	10140B	94447B	Client
83227 byte	346B	9628B	93201B	Server

From the data shown, we can see the amount of memory used. The values given are in bytes. The application obtains the amount of ROM used from the "Text+Data" data. The amount of RAM used is obtained from the value "Data+Bss." In this case, the amount of RAM used by the client is 84307 bytes, and the amount of ROM is 10562 bytes. The server uses 83573 bytes of ROM and 9974 bytes of RAM.

The handshake phase of DTLS is the most resource-intensive phase. Especially the authentication and key exchange phases consume significant resources. The handshake phase consumes significantly different amounts of memory in symmetric and asymmetric encryption. Table 4 shows the amount of memory consumed in the handshake phase.

Table 4. Memory Consumption in Elliptic Curve and Symmetric Encryption Modes

Phase	Server	Client	Total
DTLS-SE	69B	129B	198B
DTLS-ECE	408B	441B	849B

Even if the elliptic curve method is used in asymmetric encryption, it significantly consumes memory and burdens the processor. It poses a significant problem, especially for Class-0 and Class-1 devices.

6.4. Energy Consumption

The Powertrace library was used to calculate the power consumption of the application implemented in the simulation environment. Using the relevant method of this library, the CPU cycles of the Wismote nodes at 10-second intervals and their corresponding energy consumption were obtained, as shown in Tables 5 and 6, respectively.

Table 5. Number of Processor Cycles Obtained from Nodes with 10 s Interval

CPU	LPM	TX	RX
223	196.212	270	126.677
1.208	818.580	1.425	752.565
2.046	1.375.614	2.375	1.308.646
2.998	2.013.722	3.394	1.865.354
4.203	2.812.932	4.755	2.743.367
5.357	3.603.987	6.112	3.533.275
6.713	4.521.397	7.607	4.449.186

Table 6. The amount of energy consumed by the Wismote node in mW

CPU	LPM	TX	RX	Toplam
0,00128069551	0,80920193816	0,00147572525	0,81377863687	1,6257369957
0,00108956634	0,72425476956	0,00123518857	0,72301568039	1,449565912
0,00123778896	0,8296670625	0,00132490226	0,72383090048	1,5560606542
0,00272521604	0,8285285222	0,00176957015	1,14159118292	1,9745824267
0,00156673918	0,8391316407	0,00176436936	1,02703719435	1,8694999435
0,00150042906	1,19281510301	0,00194379673	1,19088610556	1,9871454343

The CPU value shown in Table 5 indicates the total CPU cycles; LPM indicates the total number of cycles in low power mode; TX indicates the total number of cycles in the transmit state; and RX indicates the total number of cycles in the receive state. The power consumption value in mW is calculated using the formula given below.

The values in Table 6 are calculated using the formula in (1). The energy value given in the formula is the cycle difference between the two runs, and the voltage value is the voltage (3 volts) at which the node is powered. The RTIMER value was obtained with the method added to the code and is 34962. The Wismote node consumes an average energy of 1,6 mW.

$$Energy\ Value = \frac{Energy_{est} Value * Voltage\ Value}{RTIMER_{SECOND} * Execution\ Time} \quad (1)$$

Asymmetric encryption significantly increases not only the code size but also the number of messages exchanged during the DTLS phase. However, since elliptic curve encryption significantly reduces the code size compared to other asymmetric encryption methods, the idea of using it on resource-constrained nodes has emerged, and studies have focused on this. However, elliptic curve encryption proves ineffective on Class-0 and Class-1 devices, even when used. Even the TinyECC code (Liu & Ning, 2008), one of the most important studies on this subject, was not run on Sky and Z1 nodes but on the Wismote node by throttling various resources. Compared to other studies, the TinyDTLS code (Bergmann, 2017) seems to work effectively. However, the Contiki operating system and other protocols use a significant amount of code, highlighting the

need for a more efficient elliptic curve method. Especially for Class-0 devices, optimizing the code is necessary to provide end-to-end encrypted communication. Figure 9 shows the comparison of the studies according to the amount of RAM and Rom used.

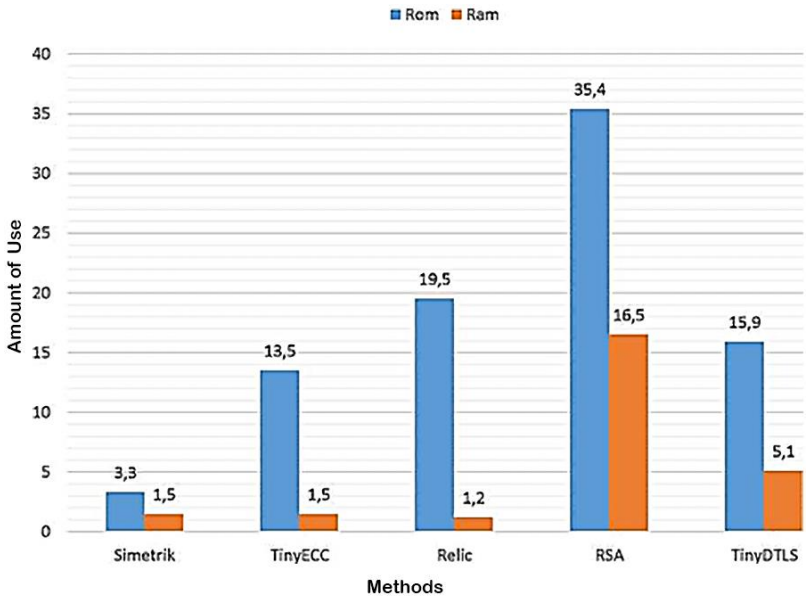

Figure 9. RAM and ROM Usage of Elliptic Curve Studies with SC and RSA

Although elliptic curve encryption reduces resource consumption compared to other asymmetric encryption methods, symmetric encryption consumes much fewer resources. Especially when compared to RSA, there is a large amount of gain. However, using asymmetric encryption is necessary for the communication of devices connected to the Internet of Things, which is inherent in the Internet of Things. In addition, as the structure grows, the disadvantage of symmetric encryption due to the number of keys becomes apparent. In an environment with N nodes, symmetric encryption requires a total of N (N-1) keys for each node to communicate with other nodes. In asymmetric encryption, each node has a public and a secret key, and the total number of keys is 2N. This shows how difficult it is to use symmetric encryption when the number of nodes reaches millions. The researchers found that the TinyECC library was the most effective among the elliptic curve encryption methods used. It has been shown that the elliptic curve encryption method, which is considerably faster and less resource-consuming than other asymmetric encryption methods, can be applied to resource-constrained devices if it is emphasized.

7. CONCLUSION AND FUTURE WORK

In this study, secure communication is realized using the elliptic curve encryption method at the DTLS layer in the nodes to provide end-to-end communication on resource-constrained devices in the Internet of Things in the Cooja simulation environment. The SHA2 hashing algorithm is used to ensure data integrity. Even if someone intercepts the data, it will be meaningless without a key. In addition, the hash algorithm provides protection against data modification and replay attacks. In this way, it is shown that elliptic curve encryption can be used on resource-constrained devices. Simulation results are presented, and it is shown that asymmetric encryption does not work effectively on resource-constrained devices, especially on Class-0 and Class-1 devices. In this study, the Sky node and Z1 node, which have severe resource constraints, could not be used for the reasons mentioned, and the Wismote node was preferred. However, in the Internet of Things, it is seen as a necessity to integrate asymmetric encryption methods in order for devices to be connected to the Internet at any time and to communicate end-to-end with other devices. For this

reason, studies have shown that elliptic curve encryption imposes a serious burden on wireless sensor devices unless optimization is performed.

Comparisons of memory usage were made for some of the developed libraries. The amount of memory used by some of the developed libraries was compared. The TinyECC library was found to be the most efficient elliptic curve encryption method, and its dynamic structure allows for the desired optimizations to be made. In future studies, some improvements to the method are aimed at reducing the energy consumption of resource-constrained sensors.

Contribution of Authors
Authors contributed equally to the article.

Conflict of Interest Statement
There is no conflict of interest between the authors.

Statement of Research and Publication Ethics
The study complied with research and publication ethics.

REFERENCES

Aazam, M., St-Hilaire, M., Lung, C.-H., & Lambadaris, I. (2016). PRE-Fog: IoT trace based probabilistic resource estimation at Fog. *13th IEEE Annual Consumer Communications & Networking Conference (CCNC).*

Aloul, F., Zualkernan, I., & Mahmoud, R. (2015). Internet of things (IoT) security: Current status, challenges and prospective measures. *10th International Conference for Internet Technology and Secured Transactions (ICITST)*, (336-341). Londra.

Bergmann, O. (2017). *Eclipse tinydtls.* Retrieved February 10, 2024 from Eclipse Foundation: https://projects.eclipse.org/projects/iot.tinydtls

Bormann, C., Ersue, M., & Keranen, A. (2014). *Terminology for Constrained-Node Networks.* Internet Engineering Task Force (IETF).

Bozkurt, Ö. (2005). *Eliptik Eğri Şifreleme Kullanarak Güvenli Soket Katmanı Protokolünün Gerçeklemesi ve Performansının Değerlendirilmesi.* [Yüksek Lisans Tezi]. İstanbul: Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü, İstanbul.

Chavan, A., & Nighot, M. (2014). Secure CoAP Using Enhanced DTLS for Internet of Things. *International Journal of Innovative Research in Computer and Communication Engineering*. 78, 646-651.

Chen, X. (2014). *Constrained Application Protocol for Internet of Things.* Retrieved February 10, 2024 from https://www.cse.wustl.edu/~jain/cse574-14/ftp/coap.pdf

Franco, J. (2024). *20-CS-6053 - Network Security.* Retrieved February 10, 2024 from University of Cincinnati Electrical Engineering & Computer Science: http://gauss.ececs.uc.edu/Courses/c6053/lectures/PDF/elliptic.pdf

Görmüş, S., Aydın, H., & Ulutaş, G. (2017). Nesnelerin interneti teknolojisi için güveenlik: var olan mekanizmalar, protokoller ve yaşanılan zorlukların araştırılması. *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 33(4), 1247-1272.

Huawei. (2023). *Edge Networking*. Retrieved February 10, 2024 from Huawei Community Forums: https://forum.huawei.com/enterprise/en/edge-networking/thread/690495115774279680-667213860102352896

Karataş, İ., & Bayraklı, S. (2020). Contiki İşletim Sisteminde Cooja Simulatörü Kullanılarak Örnek Bir Nesnelerin İnterneti Uygulaması. *Avrupa Bilim ve Teknoloji Dergisi*, 19, 763-769.

Li, S., & Xu, L. (2017). *Securing the Internet of Things.* Syngress.

Liu, A., & Ning, P. (2008). TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks. *7th International Conference on Information Processing in Sensor Networks (IPSN'08).* Washington DC.

McCumber, J. (2004). *Assessing and Managing Security Risk in IT Systems.* Auerbach Publications.

Microsoft. (2021, 07 04). *Chapter 3: Functional description of Azure RTOS NetX Secure DTLS*. Retrieved February 10, 2024 from Microsoft Learn: https://learn.microsoft.com/en-us/azure/rtos/netx-duo/netx-secure-dtls/chapter3

Nakagawa, I., & Shimojo, S. (2017). IoT Agent Platform Mechanism with Transparent Cloud Computing Framework for Improving IoT Security. *IEEE 41st Annual Computer Software and Applications Conference (COMPSAC).*

Orhon, N. (2015, 06 17). *Computer Engineering Department Seminar #60*. Retrieved February 10, 2024 from Yaşar Üniversitesi Bilgisayar Mühendisliği Bölümü: https://ce.yasar.edu.tr/en/2015/06/neriman-gamze-orhon-elliptic-curve-cryptography-and-efficient-implementations-june-17th-2015-friday/

Raza, S., Shafagh, H., Hewage, K., Hummen, R., & Voigt, T. (2013). Lithe: Lightweight Secure CoAP for the Internet of Things. *IEEE Sensors Journal*, 3711-3720.

Rescorla, E., Tschofenig, H., & Modadugu, N. (2022). *The Datagram Transport Layer Security (DTLS) Protocol Version 1.3.* Internet Engineering Task Force (IETF).

Santos, G., Guimaraes, V., Rodrigues, G., Granville, L., & Tarouco, L. (2015). A DTLS-based Security Architecture for the Internet of Things. *20th IEEE Symposium on Computers and Communication (ISCC)*, 809-815.

Shelby, Z., Hartke, K., & Bormann, C. (2014). *The Constrained Application Protocol (CoAP).* Internet Engineering Task Force (IETF).

Stinson, D. (2005). *Cryptography: Theory and Practice.* Chapman and Hall/CRC.

Szczechowiak, P., Oliviera, L., Collier, M., & Dahab, R. (2008). NanoECC: Testing the Limits of Elliptic Curve Cryptography. *Sensor Networks*, 305-320.

Trappe, W., & Washington, L. (2005). *Introduction to Cryptography with Coding Theory*. Pearson.

Zhao, K., & Ge, L. (2013). A Survey on the Internet of Things Security. *9th International Conference on Computational Intelligence and Security*, (663-667). Leshan.