Ci task | GDAŃSK UNIVERSITY OF TECHNOLOGY

TASK Quarterly 26 (3) 2022

Lightweight cryptographic algorithm based on trigonometry, dedicated on encryption of short messages

Wiesław Maleszewski
Lomza State University of Applied Sciences
14 Akademicka, 18-400 Lomza, Poland

Date: 14 December 2023

https://doi.org/10.34808/ksxc-hn17

Abstract

The IoT technology is currently used in many areas and marked by growing popularity. On the one hand, the IoT makes our lives easier, on the other hand, it presents challenges in terms of security and privacy protection. An IoT infrastructure is characterized by a high level of threats due to, *inter alia*, numerous technical barriers that make it difficult to use conventional methods to protect information. The aim of this paper is to present a symmetric coding algorithm based on algebraic groups generated by specific trigonometric curves. The algorithm is dedicated to short data sequences transmitted by devices with limited computing power.

Keywords:

IoT cryptography, lightweight cryptography, flexible cryptography

1. Introduction

The term “Internet of Things” (IoT) was coined in 1999 by Kevin Ashton to propose a system which would involve real objects in the physical world and could be linked to the network via sensors. Additionally, the term was used to describe the capability of linking RFID tags in business supply chains to the Internet to automatically count and monitor objects without human interference. However, the precise definition of the IoT is still in formation and depends on the adopted perspective [1].

IoT is a relatively recent name. The concept of connecting machines and networks to track and manage objects dates back decades. By the end of the 1970s, technologies for tracking individual electrical grid meters over telephone lines were widely used. However, it was in the early 90s that substantial developments in the wireless sensing technology enabled a widespread adoption of ’machine-to-machine’ (M2M) corporate and industrial systems for monitoring and operating equipment.

The term ’Internet of Things’ (IoT) has gained prominence in recent years due to the increasing use of smart devices that describe everyday objects. It has made it possible to alter our daily lives in completely new ways that are more automated, effective, and practical. The IoT communication serves as a foundational element of the IoT environment. However, concerns about privacy and security are not well handled because of the public nature of wireless communication in sensing devices and also the limited capacity comprising storage, bandwidth, and energy. In IoT communication, it is susceptible to hacker attacks. Therefore, resolving security and privacy issues in the IoT communication is critical in an IoT environment.

The IoT is specifically characterized as a dynamic global network with automated capacities entirely based on conventions and rules [2]. The IoT can be considered a group of connected devices that can talk to each other using near-field communication (NFC) techniques. The European Telecommunications Standards Institute (ETSI) views IoT as a superset of linking machines distinctively recognized by the current NFC technology. Van Kranenburg [3] refers to the IoT as a globally interconnected network dependent on sensors, transmission, connectivity, and other processing technologies, which may represent the next generation of information and communications technology (ICT).

Despite the differences over the IoT concept, it has been extensively debated, and numerous institutions have speedily created a related technology. Specifically, smart sensing and wireless-based technologies have been included in the Internet of Things, and new problems and research areas have evolved [4]. Frenken et al. [5] reviewed IoT-based technologies, potential markets, and growing concerns.

The IoT is considered the next generation of the web, in which physical objects may be tracked and monitored over the network. Moreover, the IoT is defined differently depending on the technologies used for implementation. Nevertheless, the IoT fundamentals imply that objects in an IoT can be distinctively identifiable in their virtual representations. Within an IoT, all objects can communicate data and, if necessary, process data based on predetermined schemas.

It is anticipated by Hasan [6] that the total number of M2M connections will increase to 27 billion in 2024. This increase in the number establishes the IoT as one of the most promising emerging businesses that could serve as a foundation of the growing digital economy.

M2M connections encompass many applications, including the smart infrastructure, smart metering, security and others. Following is a brief overview of some of the many application areas of the IoT.

- **Smart infrastructures** make considerable use of the developing computing and communication resources to enhance the quality of life for the general population [7]. It comprises smart housing, smart traffic control, disaster control systems, Intelligent services, and others. Additionally, effort is made to make cities smarter, and governments throughout the world are incentivizing their growth with various incentives.
- **Smart metering** covers applications for diverse measurements, monitoring, and administration. Smart grids are the most widespread deployment of smart metering, where power demand is tracked and monitored. Additionally, smart metering can be utilized to mitigate the problems of electricity theft [8].
- **Smart monitoring** Various IoT applications are also used in the critical field of security and emergencies. It includes applications such as restricting access to restricted areas to authorized personnel only. Another utilization in this field is the detection of toxic gas leaks in corporate or surrounding areas of chemical plants. It is possible to implement security software to safeguard confidential materials and items. IoT solutions that detect various liquids can also be applied in such delicate structures to minimize corrosion and malfunctions.

The IoT devices are anticipated to be linked to the network and other devices and easily transmit directly with other network-associated devices. Following are the future developments of the IoT.

- An IoT application should be able to manage real-time data and interface with other sensors. Furthermore, it should deal with perceiving, acting, and collaborating with humans.

- In the future, some of the IoT-based prospective application categories will be based on IoT standpoints. This application could be used to forecast natural disasters. Simulations of the performance of different phenomena are used in commercial processes.
- Water security tracking applications, applications required for an intelligent housing design, and healthcare applications that involve monitoring activities and health factors [9], as well as health inputs.
- Other agricultural uses include sophisticated packing, text message warnings regarding land defects, and intakes. More on future uses: intelligent transportation system design applications such as vehicle tracking, law enforcement, and pollution management.
- Some potential developments will also focus on the green infrastructure and smart security. Additionally, many businesses focus on the development of IoT applications, while some focus on IoT environments.

Nevertheless, as more and more devices become connected to the Internet, the cyber-physical environment becomes more vulnerable to network attacks. They include assaults on the network connectivity and service integrity, including the network confidentiality and identity. The attackers commit data theft, denial of service, and service corruption by changing data [10, 11, 12].

Subsequent generations of applications across a wide range of domains, including (but not limited to) mechanization, computer science, telecommunication and e-health, including engineering, will be significantly impacted by the Internet of Things (IoT), which is evolving into a pervasive paradigm. The growing amount of research and development being done on commercial models and procedures, as well as the upcoming next-generation advancements in a variety of fields, such as 5G [12], autonomous systems, and wireless low-power communication, show the significance of the IoT. The planning, development, and administration of systems for IoT applications present several issues, one of which calls for technologies that may transform IoT into a green as well as energy-efficient paradigm. In reality, enhanced information and communication and collaboration capabilities as well as their sustainable development and power efficiency will be necessary for the enslavement of wireless sensor nodes, fully independent systems (robots, vehicles, UAVs), machine-to-machine, medical and industrial IoT, or other related technologies, particularly for large-scale implementations of IoT applications [13, 14].

2. Sensor networks

The recent advancements in wireless communications and technology have facilitated the development of minimal-cost, energy-efficient, multipurpose sensor nodes that are compact, yet transmit data wirelessly over short distances [15]. Furthermore, sensor networks power these miniature sensor nodes, consisting of sensing, data analysis, and communication components. As a result, sensor networks offer substantial advantages, compared to regular sensors.

A sensor network comprises a significant number of sensor nodes that are extensively positioned inside or extremely close to the event. Sensor node positions do not need to be developed or predefined. This enables uncontrolled deployment in remote areas [16]. On the other hand, this implies that sensor-based conventions and methods must be self-organizing. Moreover, the collaborative approach to sensor nodes is another distinguishing aspect of sensor networks. These sensor nodes include an inbuilt processor. Rather than transmitting raw data to multi-fused nodes, they employ their processing capacity to perform internal computations and transfer only the appropriate and partially filtered data.

The characteristics outlined above ensure that sensor networks can be used for a wide variety of applications which include medicare, defense and home application areas.

Over the past few years, with the progress of standard technologies, wireless sensor networks have significantly emerged into new domains. The emphasis is on building novel transmission standards and control services to address sensor node-specific requirements including minimal energy, processing capability, and storage. In addition, topology generation, control, and management are some of the hot research areas in WSNs. Tab. 1 provides the leading research developments in wireless sensor networks.

Energy efficiency is a critical concern in sensor networks because the battery capacity is inadequate. Additionally, installing nodes is the initial stage in constructing a sensor network. On the other hand, sensor nodes are battery-enabled and distributed at random in the target region. However, sensor nodes have substantial problems due to the battery power restrictions, computing power limits, unprocessed data collection, and the network limited memory. Optimizing the energy usage is a critical problem in WSNs to extend the network lifetime. Researchers have significantly emphasized this topic in recent years to resolve the issue. In a multi-hop situation, nodes near the sink provide their own data as well as data acquired by other nodes away from the sink, assuming that the sensor nodes are implemented

Table 1: Novel research developments in Wireless Sensor Networks and IoT

Research	Scope	Detail
SmartSantander (Sanchez, 2014)[17]	Smart city	The SmartSantander project pursues to provide an experimental test facility for the advancement and experimentation of city-scale designs, essential supporting state-of-the-art methods, services, and other IoT applications.
Mercury (Lorincz, 2009)[18]	Healthcare	Developing body-worn sensors to track physiological data of patients.
WiSeNts (Marron, 2006)[19]	Embedding system	WiSeNts take advantage of the existing research in integrated systems, pervasive systems, and WSNs to enable collaborating objects.
MOSAR (Rawat, 2014)[20]	Healthcare	Implementation of large-scale wireless sensor networks to acquire the dynamics of patient-patient and patient-doctor communication to combat the bacterial antibiotic resistance.

equally. In this situation, the sensor nodes closest to the sink utilize more energy and drain rapidly [21]. However, if the multi-hop is not employed and all nodes broadcast their data directly to the base station (BS), the nodes farthest away from the BS die faster than the nodes nearest to the BS since they require more communication power to deliver their data to the BS [22]. Thus, the sensor network is detached, with substantial energy remaining unused, resulting in a significant reduction in the network lifetime. Energy holes produce a network division, in such a way that a full network connectivity is impossible.

Various solutions have been introduced to tackle the Energy Hole Problem (EHP) [23], the compressed sensing approach [24], and the sink or node mobility which primarily implements replicated nodes near the sink, are some strategies that mitigate the energy hole problem. The goal of energy hole avoidance is to postpone or avoid the emergence of the energy hole in order to increase the network longevity.

IoT applications presently confront a myriad of security issues. Foremost, there exists an imperative for a cutting-edge, comprehensive framework tailored for IoT applications. Such an application is not an isolated entity but rather a culmination of collaborative endeavors from various individuals and corporations. Multiple devices and systems operate synergistically across each layer, from data acquisition to application execution. This integration involves a multitude of sensors and actuators located at the edge nodes. Owing to the vast diversity of protocols, techniques and sensors inherent in IoT applications, significant trade-offs arise among the cost-efficiency, security, reliability, privacy, responsiveness and other parameters. Tab. 2 shows some of the security challenges in IoT-based sensor networks.

The IoT is constantly evolving with the progress of the supporting technologies, i.e., wireless communication, which includes wearable sensors, RFID, wireless sensor networks, NFC, machine to machine devices, and actuators [30]

The IoT is considered an essential tool in a wide range of industries, an important area of future technology. The IoT is very valuable for enterprises. Its value can be realized when there is the ability to communicate between the connected devices and integrate with vendor-managed inventory systems, business intelligence applications, customer support systems, and business analytics. Sensory networks are considered an essential part of the IoT. They consist of dimensional distributed sensor-equipped devices to sensor environmental and physical conditions that, in cooperation with the RFID, can track the activity of things such as their temperature, location, and movements [31]. Managing the sheer number of connected things presents one of the largest challenges to the IoT device security.

Many methods can be implemented by users for their devices to be trusted by other users to protect short-term information used in the Internet of Things sensor network. There are several existing methods available which can be used to guard short-term information. It is essential for these sensor networks to make sure that the security requirements are met to safeguard data from damaging attacks and tampering. Although there are different types of security methods, they mainly consist of the password security, secure network, and application-level encryption procedures.

Password Security: is designed for protecting sensitive information such as messages and data on a secure or encrypted medium using a password code. When a system or a device requests a password, it must be provided to access the protected content.

Secure Network: is considered the most common

Table 2: Possible security challenges in IoT-based sensor networks

Challenges	Description
IoT device Privacy	IoT devices are prone to leaking sensitive user information.[25].
DoS attacks	It shuts down the system, resulting in the inability to access services [26].
Unauthorized Access	Middle-ware Layers provide diverse gateways to applications and data stores. The attacker might easily cause harm to the system by restricting access to IoT-related services or erasing existing data [27].
Eavesdropping	During packet forwarding, attackers may eavesdrop on transmissions [28].
Sniffing Attacks	Sniffer programs may be used by attackers to monitor network data in IoT applications. If there are no adequate security mechanisms in place to prevent it, the attacker can obtain access to sensitive user data [29].
Access based Control	It is a method that permits only authorized individuals or processes to acquire data or accounts [25].

method of providing a security system to an individual and is designed with a digital signature that is generated and stored on a secure server.

Application-Level Encryption: Specially designed for protecting application-level information, with solid cryptographic techniques based on public-private key pairs [32].

The current issue related to the IoT sector is to determine how we can maintain our privacy and safety when all our daily activities are exposed online, and we can be watched by businesses, governments, factories, etc. The IoT has been an excellent tool to generate vast amounts of data used in commercial products. It can track a person's location, movements, activities and purchasing habits. There are different methods available in the IoT market to protect data. Both the password security and a secure network can be considered the most common methods of providing a security system to an individual. The application-level encryption serves as an excellent solution for the institute which has relatively less storage space for storing sensitive data without any complications that may be caused by the other above mentioned methods.

3. Popular cryptographic algorithms

Cryptographic algorithms are used to secure everything from web browsing to banking transactions. The damage could be devastating if one falls prey to a hacker or a data breach. These days, there are numerous types of cryptography. The most popular include symmetric key cryptography and asymmetric key cryptography; both are easier to understand, but the asymmetric key cryptography is more commonly used in various applications, as it offers a distinct advantage for secure communication techniques. Some of the most popular cryptographic algorithms are:

- Data Encryption System (DES)
- Advanced Encryption Standard (AES)
- Rivest-Shamir-Adleman (RSA)
- ElGamal cryptosystem
- Message Digest 5 (MD5)
- Secure Hashing Algorithms (SHA)

The Data Encryption Standard (DES) is a symmetric block cipher designed in 1975 by IBM for the then US National Bureau of Standards (NBS) [33], presently the National Institute of Standards and Technology (NIST). From 1976 to 2001, the DES was the U.S. federal standard, and from 1981 the ANSI standard for the private sector. For the past decade, the DES has been recognized as an algorithm that does not provide adequate security, mainly due to its short key length, making it very vulnerable to a brute force attack [34].

The successor to the DES algorithm is the Advanced Encryption Standard (AES) [35]. The AES encryption is arguably the predominant method for safeguarding data at rest. It is employed in various domains, including self-encrypting drives, database encryption, and storage encryption. Over the years, AES has seen extensive implementation across different platforms and libraries, including, but not limited to Libgcrypt, wolfSSL, GnuTLS, Network Security Services, OpenSSL, LibreSSL, embed TLS, axTLS, Microsoft CryptoAPI, tiny-AES-c, Solaris Cryptographic Framework, OpenAES, LibTomCrypt, libSodium, AES Dust, Crypto++, Java Cryptography Extension (available in the Java Runtime Environment from version 1.4.2), PyCrypto, and AES-JS.

In the realm of the archiving and compression tools, the AES is utilized by solutions such as 7z, Amanda Backup, PeaZip, PKZIP, RAR, WinZip, and UltraISO. Moreover, the majority of encrypted file systems, like NTFS, adopt the AES. The disk encryption software, encompassing BitLocker, CipherShed, DiskCryptor, FileVault (integrated with Mac OS X), GBDE, Geli, LibreCrypt, LUKS, Private Disk, TrueCrypt, and VeraCrypt, also leverage the AES algorithm for security.

Rivest-Shamir-Adleman (RSA) and ElGamal asymmetric algorithms are both block cipher algorithms, often called the 'block cipher' that allow storage of large amounts of data on small memory devices such as smart cards. There are two main differences between the algorithms: the size of their keys and the way in which they handle random numbers. While both share a standard security level, some experts have identified theoretical vulnerabilities in the RSA algorithm that may not exist in elliptic curve cryptography. However, these have never been exploited in practice. This is a result of its smaller key sizes, which make it easier for a hacker to find ways to break it. When installed on your computer, RSA generates a public/private key pair (consisting of two numbers). The RSA Key Generator software takes randomly generated prime numbers as inputs, multiplies them together, and then performs an integer modulus operation to produce two public/private key pairs (each consisting of two values). Elgamal Asymmetric Algorithms are found to be relatively fast over the cloud network and slow in a local system. However, the Rivest-Shamir-Adleman (RSA) algorithm is very slow on both the cloud network and the local system. RSA needs better and more robust configuration machines with a more significant number of processors, more ROM and cache memory, and fast processors compared to other cryptographic algorithms for its fast operation from the cloud network [36].

MD5 is used both on its own and as a collision-resistance primitive in the asymmetric cryptography families: Digital Signature Schemes (DSA), also known as ElGamal, and the elliptic curve-based cryptography (ECC) [37]. SHA is used by itself, as a hash function, in Message Authentication Codes. MD5 and SHA are widely accepted as 'cryptographic hashes' of file data. As it was initially developed for the print media, MD5 can be used to digitally sign files and verify that they come from their claimed authors. The security of the MD5 hash function is based on its low collision rate and related tests. SHA is used in a much more comprehensive range of roles. It uses six standard hashing functions folded into one algorithm, combining hashes to create a larger hash output. This means it can be used as a hash function, as a checksum function, or even as a Hash-Based Message Authentication Code (HMAC), and since it is included in so many other cryptographic algorithms, most modern security protocols use SHA. The security behind SHA is derived from the size of the output values; if an attacker can find two inputs with the same hash value, then they could re-use that to forge data signatures and fool the authentication schemes. There are many different iterations of SHA, but all use the same essential hashing function, which means that the two most important properties of SHA, collision resistance and one-Waynes, are present [38].

The methods for evaluating the security of cryptographic algorithms are of two general kinds: theoretical cryptanalysis and mathematical cryptanalysis. In the theoretical cryptanalysis, the goal is to show that a cryptographic algorithm is flawed in some way; in the mathematical cryptanalysis, the goal is to show that there exist significantly faster algorithms for performing a given task than a given cryptographic one [39].

A lot of the Internet of Things sensors collect and transmit data with a simple structure, the recording of which takes only several or a dozen bits. The use of the methods described above requires augmenting the transmitted information with noise and then encoding it. Such a procedure generates additional costs, and its use adversely affects the functionality of power-limited devices.

Below will be presented an encryption algorithm with a very flexible block structure. These algorithms for the protection of very short information can work in blocks of minimal length, while if there is a need to encrypt longer information or increase the level of security, it is necessary to use more extensive blocks.

4. A cryptosystem inspired by the topologist's sine curve

Many popular cryptographic algorithms have been inspired by geometric structures. The topologist's sine curve [40] is a set of points:

$$T = \left\{ \left(x, \sin\frac{1}{x} \right) : x \in (0,1] \right\} \cup \{ x = 0 \wedge y \in [-1,1] \} \quad (1)$$

Let us consider the arithmetic sequence x_t, in the form:

$$x_t = x_1 + (t-1)r, \quad (2)$$

where x_1 and r are arbitrarily small positive real numbers and $t \in \{0,1,2,\ldots\}$, then with any arbitrary, but fixed $k \in \mathbb{N}$, we can define the function:

$$f_k : \mathbb{R}_+ \cup \{0\} \to \mathbb{N}, \quad (3)$$

given by the formula:

$$f_k(x_t) := \begin{cases} \left\lfloor k \sin\frac{1}{x_t} \right\rfloor + k + 1 & \text{when} \quad x \neq 0 \\ 0 & \text{when} \quad x = 0 \end{cases}, \quad (4)$$

the domain and set of values which consists of a discrete set of non-negative numbers [41]. Then let us consider the collection:

$$G_{2k} = \bigcup_{i=1}^{2k} \{f_k(\widetilde{x}_i)\}, \tag{5}$$

where:

$$\widetilde{x}_1 = x_1, \tag{6}$$

and

$$\widetilde{x}_i := \min\left\{ x_i : \mathop{\forall}\limits_{j=1,2,\dots,i-1} f_k(\widetilde{x}_j) \neq f_k(x_i) \right\}. \tag{7}$$

The set G_{2k} contains $2k$ of the initial different values of the function f_k defined on the sequence x_t. In the set $G = G_{2k} \cup \{0\}$ we define the mapping:

$$+_k^f : G \times G \to G, \tag{8}$$

specified using the formula:

$$f_k(\widetilde{x}_i) +_k^f f_k(\widetilde{x}_j) := f_k\Big((\widetilde{x}_i + \widetilde{x}_j) \pmod{2k+1}\Big). \tag{9}$$

Let us consider the sequence $x_t = 10^{-6}t$ where $k = 1, 2, 3, \dots$ and the function:

$$f_2(x_t) = \left\lfloor 2 \sin \frac{1}{x_t} \right\rfloor + 3. \tag{10}$$

Table 3: The laws of arithmetic in $(G, +_2^{f^*}, 0)$ for $x_0 = r = 10^{-6}$

$+_2^{f^*}$	0	1	2	3	4
0	0	1	2	3	4
1	1	2	4	0	3
2	2	4	3	1	0
3	3	0	1	4	2
4	4	3	0	2	1

In the following work, we will name such tables as 'encryption tables' and mark them as: $T_{2k+1}(x_1, r)$.

A permutation is any arrangement of all the elements of a set in the order. Permutation π of $2k+1$ elements set is function:

$$\pi : \{0, \dots, 2k\} \to \{0, \dots, 2k\} \tag{11}$$

represented in a two-line form by:

$$\begin{pmatrix} 0 & 1 & \cdots & 2k \\ \pi(0) & \pi(1) & \cdots & \pi(2k) \end{pmatrix} \tag{12}$$

The permutation matrix $\pi = (p_{ij})$ obtained by permuting the columns of the identity matrix I_{2k+1}, will be referred to as the column representation:

$$P_\pi = \begin{bmatrix} \mathbf{e}_{\pi(0)} \\ \mathbf{e}_{\pi(1)} \\ \vdots \\ \mathbf{e}_{\pi(2k)} \end{bmatrix} \tag{13}$$

where e_j, a standard basis vector, denotes a row vector of length $2k+1$ with 1 in position $j+1$ and 0 in every other position. For example, the permutation matrix P_{π_1} corresponding to the permutation

$$\pi_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 0 & 2 \end{pmatrix} \tag{14}$$

is

$$P_{\pi_1} = \begin{bmatrix} \mathbf{e}_{\pi(0)} \\ \mathbf{e}_{\pi(1)} \\ \mathbf{e}_{\pi(2)} \\ \mathbf{e}_{\pi(3)} \\ \mathbf{e}_{\pi(4)} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_4 \\ \mathbf{e}_3 \\ \mathbf{e}_0 \\ \mathbf{e}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{15}$$

Moreover, there is another representation which is determined by the permutation of the rows of the identity matrix I_{2k+1}, in this representation [42], thus:

$$\forall_{i,j \in \{0,\dots,2k\}} p_{ij} = 1 \Leftrightarrow i = \pi(j) \tag{16}$$

Multiplying by P_π a column of vector g will permute the rows of the vector:

$$P_\pi \mathbf{g} = \begin{bmatrix} \mathbf{e}_{\pi(0)} \\ \mathbf{e}_{\pi(1)} \\ \vdots \\ \mathbf{e}_{\pi(2k)} \end{bmatrix} \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{2k} \end{bmatrix} = \begin{bmatrix} g_{\pi(0)} \\ g_{\pi(1)} \\ \vdots \\ g_{\pi(2k)} \end{bmatrix} \tag{17}$$

Let us establish that we are using an input array of size $(2k+1)$ by $(2k+1)$. The consequence of this assumption is that we use blocks that consist of $(2k+1)$ elements. The suggested encryption algorithm consists of three stages. The first stage involves obtaining a message length that is a multiple of the length of a single block. If $n \equiv 0 \bmod (2k+1)$ then we split the message into blocks. Otherwise if $m \equiv l \bmod (2k+1)$ then we complete the message in a certain way to a full block before splitting, i.e.:

$$M = (\underbrace{\underbrace{M_1, M_2, \dots M_n}_{n}, M_1, M_2, \dots, M_{2k+1-l}}_{c(2k+1)}) \tag{18}$$

By splitting the messages, we get t blocks of $2k+1$ elements. Writing the marks in a system compatible with the American Standard Code for Information Exchange, we can assume that the plaintext is represented by a matrix of the following type:

Table 4: Permutations read from the appropriate rows

$k_1 = 3$	$\pi_1 = (3,0,1,4,2) = (0,1,4,2,3)$	$\pi_1^{-1} = (2,4,1,0,3) = (0,3,2,4,1)$
$k_2 = 0$	$\pi_2 = (0,1,2,3,4)$	$\pi_2^{-1} = (4,3,2,1,0) = (0,4,3,2,1)$
$k_3 = 1$	$\pi_3 = (1,2,4,0,3) = (0,3,1,2,4)$	$\pi_3^{-1} = (3,0,4,2,1) = (0,4,2,1,3)$
$k_4 = 4$	$\pi_4 = (4,3,0,2,1) = (0,2,1,4,3)$	$\pi_4^{-1} = (1,2,0,3,4) = (0,3,4,1,2)$
$k_5 = 2$	$\pi_5 = (2,4,3,1,0) = (0,2,4,3,1)$	$\pi_5^{-1} = (0,1,3,4,2)$

Table 5: For ease of reading, the permutations are presented in a two-line notation

$k_1 = 3$	$\pi_1 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 4 & 3 & 0 & 2 \end{pmatrix}$	$\pi_1^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 0 & 4 & 2 & 1 \end{pmatrix}$
$k_2 = 0$	$\pi_2 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \end{pmatrix}$	$\pi_2^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 0 & 1 & 2 & 3 \end{pmatrix}$
$k_3 = 1$	$\pi_3 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 2 & 4 & 1 & 0 \end{pmatrix}$	$\pi_3^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 4 & 3 & 1 & 0 & 2 \end{pmatrix}$
$k_4 = 4$	$\pi_4 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 0 & 3 \end{pmatrix}$	$\pi_1^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 3 & 2 & 0 & 4 & 1 \end{pmatrix}$
$k_5 = 2$	$\pi_5 = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 0 & 4 & 1 & 3 \end{pmatrix}$	$\pi_1^{-1} = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 3 & 0 & 4 & 2 \end{pmatrix}$

$$M = \begin{bmatrix} \mu_{11} & \mu_{12} & \cdots & \mu_{1\ 2k+1} \\ \mu_{21} & \mu_{22} & \cdots & \mu_{2\ 2k+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{t1} & \mu_{t2} & \cdots & \mu_{t\ 2k+1} \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_t \end{bmatrix} \tag{19}$$

where

$$\mu_{ij} = \left(M_{(i-1)(2k+1)+j}\right)_2 . \tag{20}$$

The following describes the process of encrypting a single block. The second stage consists of a series of rounds, and in each round we base the process on the matrix created from the encryption table.

In addition to the encryption table, we also need a key. In the suggested algorithm, we assume that the key is the number K. Based on the key and the construction of the encryption table, we create the keys of the rounds k_i, which are the elements of the row of this table as defined by the number K, e.g.

$$K = 3 \rightarrow [k_1, k_2, k_3, k_4, k_5] = [3, 0, 1, 4, 2] \tag{21}$$

The encryption matrix, identified as $T_{(2n+1)}(x_1, r)$ is formed by taking rows from this table according to the order of each round key.

$$T_{(2n+1)}(x_1, r) = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1\ 2k+1} \\ a_{21} & a_{22} & \cdots & a_{2\ 2k+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{2k+1\ 1} & a_{2k+1\ 2} & \cdots & a_{2k+1\ 2k+1} \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_{2k+1} \end{bmatrix} \tag{22}$$

where r_i is the i-th row of the matrix $T_{(2k+1)}(x_1, r)$.

Back to the example. The key of the first round k_1 was the number 3, then the first row of the newly generated matrix becomes the third row of the table. Similarly, the second key of the second round was the number $k_2 = 0$, then the second row of the encryption matrix becomes the zero row of the table. In this way we get an encryption matrix of the following form:

$$T_5(10^{-6}, 10^{-6}) = \begin{bmatrix} 3 & 0 & 1 & 4 & 2 \\ 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 4 & 0 & 3 \\ 4 & 3 & 0 & 2 & 1 \\ 2 & 4 & 3 & 1 & 0 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \\ r_5 \end{bmatrix} \tag{23}$$

The encryption algorithm as presented is block-based. In this algorithm, we encrypt each block by an analogous process. The following describes the encryption work for the first block represented by the first row of the matrix M:

In the first step, we act with the XOR function on the first block from the message and row r_1 of the Tab. 3:

$$\begin{aligned} (\mu_1)_2 \oplus (r_1)_2 = \\ = [(\mu_{11})_2 \cdots (\mu_{1\ 2n+1})_2] \oplus [(a_{11})_2 \cdots (a_{1\ 2n+1})_2] = \\ = [(\mu_{11})_2 \oplus (a_{11})_2 \cdots (\mu_{1\ 2n+1})_2 \oplus (a_{1\ 2n+1})_2] \end{aligned} \tag{24}$$

Then we rearrange the characters of this message according to the π_1 permutation:

$$(c_{1,1})_2 = \pi_1 \left((\mu_1)_2 \oplus (r_1)_2\right), \tag{25}$$

proceed similarly:

$$(c_{1,2})_2 = \pi_2 \left((c_{1,1})_2 \oplus (r_2)_2\right), \tag{26}$$

and after m steps we have the following form:

$$(c_{1,m})_2 = \pi_i \left((c_{1,m-1})_2 \oplus (r_m)_2\right). \tag{27}$$

The number of rounds carried out with this method should depend on the length of the encryption block. This is a parameter that can also be a part of the master key. This concept will be described later in this article. In the last round, a designed operation to differentiate the encryption characters matching the same plaintext characters is performed. The effect of this operation is to introduce a state in which not only the same plain text letters can come on different ciphertext letters, but also different plaintext letters can come on the same ciphertext letters.

The α parameter is intended to disperse the ciphertext characters into ASCII table characters.

$$c_1 = (ASCII\left[[c_{1,i}]_{10} + \alpha[r_1]_{10} - 32[1,1,\dots,1]\right] \bmod 94) + 32[1,\dots,1] \tag{28}$$

By proceeding in the same way with the other blocks, we get the ciphertext of the message

$$C = (c_1, c_2, \dots c_t). \tag{29}$$

The proposed algorithm is a symmetric cipher, hence decryption is similar to encryption, however, all operations are performed in the reverse order. Having a message in an encrypted form that we want to decrypt, first we divide it into blocks. In this way, we receive a message in the form (29). In the particular message blocks, we perform the decryption process according to the procedure described below for the first block.

In the decryption process, in the first step we perform the reverse operation to the last operation that was performed in the encryption process.

$$c_{1,m} = ([[c_1]_{10} - \alpha[r_1]_{10} - 32[1,1,\dots,1]] \bmod 94) + 32[1,\dots,1] \tag{30}$$

In the next step, we connect the message to the individual keys of the rounds using the XOR function, starting from the end. Then, we give the result of these operations to the corresponding inverse permutation:

$$(c_{1,m-1})_2 = \pi_i^{-1}\left((c_{1,m})_2 \oplus (r_m)_2\right). \tag{31}$$

$$(\mu_1)_2 = \pi_1^{-1}\left((c_{1,1})_2 \oplus (r_1)_2\right) \tag{32}$$

In a similar way, decrypting block by block results in an plaintext message extended to a full block.

$$(\mu_1, \mu_2, \cdots, \mu_t) \tag{33}$$

After removing the overwritten elements, we get a string

$$M = (M_1, M_2, \dots M_n) \tag{34}$$

If we study the structure of the above encryption algorithm, we can see that the parameters that make up the key appear in several stages of the encryption process.

5. Scattered key

The first elements of the key are encountered in the parameters specifying the arithmetic sequence on the basis of which the group is generated. The next element of the key is the size of the group in which we perform operations. This parameter determines the size of the encryption blocks. The next element of the key is the number K which determines the order of the operations performed. The number m, which determines the number of rounds carried out in the second stage can also be considered as a key. The key component is also a scalar occurring in the last stage whose task is to disperse the results of the operation from the previous stages of encryption into the characters of the ASCII table.

In the above example, the dimensions of the encryption table were 5×5. Each digit in this table can be written on three bits in the binary system. Thus, the function carried out in the second stage of encryption performed transformations. As a result, each character could pass into eight different characters. The last stage of this encryption dispersed the message received as a result of the second stage into more possible values. When using larger encryption tables, e.g. 33×33, each number in the table is stored on six bits. The consequence of this condition is the greater power of the counter-division of transformations performed in the saved stage of encryption consisting of XOR functions and previously generated permutations. In this case, the second stage of encryption can carry out into any character in 64 other characters. As before, the third stage introduces dispersion in the selected codomain.

The above proposed algorithm can be used in various configurations. One option is to change the values of all key components in successive encryptions. Such a state requires the generation of a new encryption table each time. In order to reduce the computational costs, we can assume a certain number of rounds in which we will use the same encryption table changing only other key components such as K, m, α. In this case, we agree to a lower level of security and at the same time reduce the significant number of operations performed in the encryption process.

6. Summary

The aim of the project was to design a symmetric cipher with a high flexibility. On the one hand, this algorithm is intended to effectively encrypt short messages in devices with limited resources - where ensuring security is a significant challenge. On the other hand, with appropriately selected parameters, it is intended to ensure effective encryption of longer communications. The first two chapters are aimed to justify the need to look for new solutions due to the wide range of existing problems.

In the next stage of research, tests of the programming implementation of the described algorithm are planned. They are intended to answer a number of questions regarding the effectiveness of this algorithm

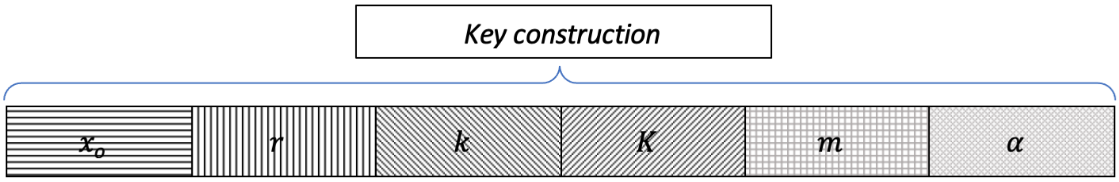

Figure 1: Key-building components

Table 6: Increase in parameter k improves the measure of dispersion of the results obtained during encoding

Parameter k	Size of encryption Tab. 3	Count of bits changed by XOR	Cardinality of codomain functions from second stage.
1	3	2	4
2	5	3	8
4	9	4	16
8	17	5	32
16	33	6	64
32	65	7	128
64	133	8	256

with various sizes of encrypted blocks. The design of the proposed algorithm allows encryption of messages of very different lengths. There are many solutions for encrypting long messages, but at the same time there is a problem with algorithms that ensure confidentiality and security of communication consisting of small amounts of data. The practice of expanding short messages into large blocks that modern solutions can handle raises the challenge of optimizing this process. This optimization makes great sense when we want to secure communication in sensor networks, which often have very limited access to power because, for example, they use built-in batteries.

In the forthcoming stages of algorithmic refinement, a meticulous examination of both its strengths and potential vulnerabilities will be imperative, particularly in the context of messages encrypted across a spectrum of key lengths. If empirical evidence suggests that the algorithm demonstrates proficiency in the encryption of concise data, yet exhibits suboptimal performance for more extended messages, the incorporation of a 'superblock' construct might warrant consideration. Within this sophisticated framework, the encryption methodologies would align with the foundational principles of the initial algorithm, albeit executed on an augmented scale. A salient challenge during the encryption of diminutive block segments will be to harmonize the interplay between block dimensions, iterative encryption rounds, the pertinence of the encryption key, and the desired security threshold.

Current research efforts focus on a rigorous comparative evaluation of the proposed algorithm with the existing cryptographic solutions presented in the previous section of the paper and with other encryption algorithms such as Ascon, LoRaWAN, ChaCha20, Zigbee, Kasumi, Salsa20 and AES. This analysis critically examines the encryption performance and the corresponding computational requirements. The results of this research will be disseminated in subsequent scientific articles.

References

[1] M. Hepp, K. Siorpaes, and D. Bachlechner, "Harvesting wiki consensus: Using wikipedia entries as vocabulary for knowledge management," *IEEE Internet Computing*, vol. 11, no. 5, pp. 54–65, 2007.

[2] D. Kiritsis, "Closed-loop plm for intelligent products in the era of the internet of things," *Computer-Aided Design*, vol. 43, no. 5, pp. 479–501, 2011.

[3] R. Van Kranenburg, "The internet of things," *World Affairs: The Journal of International Issues*, vol. 15, no. 4, pp. 126–141, 2011.

[4] D. Hunter, H. Yu, M. S. Pukish III, J. Kolbusz, and B. M. Wilamowski, "Selection of proper neural network sizes and architectures—a comparative study," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 2, pp. 228–240, 2012.

[5] T. Frenken, P. Spiess, and J. Anke, "A flexible and extensible architecture for device-level service deployment," in *European Conference on a Service-Based Internet*, pp. 230–241, Springer, 2008.

[6] M. Hasan, "State of iot 2022: Number of connected iot devices growing 18 percent to 14.4 billion globally," *IoT Analytics*, 2022. (accessed on 17 November 2022).

[7] A. Gharaibeh, M. A. Salahuddin, S. J. Hussini, A. Khreishah, I. Khalil, M. Guizani, and A. Al-Fuqaha, "Smart cities: A survey on data management, security, and enabling technologies," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2456–2501, 2017.

[8] X. Xia, Y. Xiao, and W. Liang, "Absi: An adaptive binary splitting algorithm for malicious meter inspection in smart grid," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 2, pp. 445–458, 2018.

[9] Y.-D. Lee and W.-Y. Chung, "Wireless sensor network based wearable smart shirt for ubiquitous health and activity monitoring," *Sensors and Actuators B: Chemical*, vol. 140, no. 2, pp. 390–395, 2009.

[10] J. Cichonski, J. Marron, N. Hastings, J. Ajmo, and R. Rufus, "[project description] security for iot sensor networks: Building management case study (draft)," tech. rep., National Institute of Standards and Technology, 2019.

[11] D. Choudhary, "Security challenges and countermeasures for the heterogeneity of iot applications," *Journal of Autonomous Intelligence*, vol. 1, no. 2, pp. 16–22, 2019.

[12] N. Javaid, A. Sher, H. Nasir, and N. Guizani, "Intelligence in iot-based 5g networks: Opportunities and challenges," *IEEE Communications Magazine*, vol. 56, no. 10, pp. 94–100, 2018.

[13] K. K. Patel, S. M. Patel, *et al.*, "Internet of things-iot: definition, characteristics, architecture, enabling technologies, application & future challenges," *International journal of engineering science and computing*, vol. 6, no. 5, 2016.

[14] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE wireless communications*, vol. 24, no. 3, pp. 10–16, 2017.

[15] D. Chen, J. Cong, S. Gurumani, W.-m. Hwu, K. Rupnow, and Z. Zhang, "Platform choices and design demands for iot platforms: cost, power, and performance tradeoffs," *IET Cyber-Physical Systems: Theory & Applications*, vol. 1, no. 1, pp. 70–77, 2016.

[16] M. Carlos-Mancilla, E. López-Mellado, and M. Siller, "Wireless sensor networks formation: approaches and techniques," *Journal of Sensors*, vol. 2016, 2016.

[17] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis, *et al.*, "Smartsantander: Iot experimentation over a smart city testbed," *Computer Networks*, vol. 61, pp. 217–238, 2014.

[18] K. Lorincz, B.-r. Chen, G. W. Challen, A. R. Chowdhury, S. Patel, P. Bonato, M. Welsh, *et al.*, "Mercury: a wearable sensor network platform for high-fidelity motion analysis.," in *SenSys*, vol. 9, pp. 183–196, 2009.

[19] P. J. Marron, E. W. Consortium, *et al.*, *Embedded WiSeNts research roadmap.* Logos-Verlag, 2006.

[20] P. Rawat, K. D. Singh, H. Chaouchi, and J. M. Bonnin, "Wireless sensor networks: a survey on recent developments and potential synergies," *The Journal of supercomputing*, vol. 68, no. 1, pp. 1–48, 2014.

[21] M. Perillo, Z. Cheng, and W. Heinzelman, "An analysis of strategies for mitigating the sensor network hot spot problem," in *The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services*, pp. 474–478, IEEE, 2005.

[22] R. Sharma and D. Lobiyal, "Energy holes avoiding techniques in sensor networks: A survey," *International Journal of Engineering Trends and Technology*, vol. 20, no. 4, pp. 204–208, 2015.

[23] X. Wu, G. Chen, and S. K. Das, "Avoiding energy holes in wireless sensor networks with nonuniform node distribution," *IEEE Transactions on parallel and distributed systems*, vol. 19, no. 5, pp. 710–720, 2008.

[24] V. K. Singh and M. Kumar, "A compressed sensing approach to resolve the energy hole problem in large scale wsns," *Wireless Personal Communications*, vol. 99, no. 1, pp. 185–201, 2018.

[25] M. Park, H. Oh, and K. Lee, "Security risk measurement for information leakage in iot-based smart homes from a situational awareness perspective," *Sensors*, vol. 19, no. 9, p. 2148, 2019.

[26] N. F. Syed, Z. Baig, A. Ibrahim, and C. Valli, "Denial of service attack detection through machine learning for the iot," *Journal of Information and Telecommunication*, vol. 4, no. 4, pp. 482–503, 2020.

[27] K. Riad, R. Hamza, and H. Yan, "Sensitive and energetic iot access control for managing cloud electronic health records," *IEEE Access*, vol. 7, pp. 86384–86393, 2019.

[28] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of things (iot): Taxonomy of security attacks," in *2016 3rd international conference on electronic design (ICED)*, pp. 321–326, IEEE, 2016.

[29] V. L. Narayana and A. Gopi, "Secure communication in internet of things based on packet analysis," in *Machine Intelligence and Soft Computing*, pp. 205–212, Springer, 2021.

[30] G. Cerullo, G. Mazzeo, G. Papale, B. Ragucci, and L. Sgaglione, "Iot and sensor networks security," in *Security and Resilience in Intelligent Data-Centric Systems and Communication Networks*, pp. 77–101, Elsevier, 2018.

[31] I. Lee and K. Lee, "The internet of things (iot): Applications, investments, and challenges for enterprises," *Business horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[32] R. Prabhakar, S. W. Son, C. Patrick, S. H. K. Narayanan, and M. Kandemir, "Securing disk-resident data through application level encryption," in *Fourth International IEEE Security in Storage Workshop*, pp. 46–57, IEEE, 2007.

[33] W. Diffie and M. E. Hellman, "Exhaustive cryptanalysis of the nbs data encryption standard," in *Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman*, pp. 391–414, 2022.

[34] D. E. Standard *et al.*, "Data encryption standard," *Federal Information Processing Standards Publication*, vol. 112, 1999.

[35] J. Daemen and V. Rijmen, "The advanced encryption standard process," in *The design of Rijndael*, pp. 1–8, Springer, 2002.

[36] M. Bafandehkar, S. M. Yasin, R. Mahmod, and Z. M. Hanapi, "Comparison of ecc and rsa algorithm in resource constrained devices," in *2013 International Conference on IT Convergence and Security (ICITCS)*, pp. 1–3, IEEE, 2013.

[37] N. C. Velayudhan, A. Anitha, and M. Madanan, "Sybil attack detection and secure data transmission in vanet using cmeha-dnn and md5-ecc," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2021.

[38] A. W. Appel, "Verification of a cryptographic primitive: Sha-256," *ACM Transactions on Programming Languages and Systems (TOPLAS)*, vol. 37, no. 2, pp. 1–31, 2015.

[39] M. Ahmad, M. N. Doja, and M. M. S. Beg, "Security analysis and enhancements of an image cryptosystem based on hyperchaotic system," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 1, pp. 77–85, 2021.

[40] J. J. Dijkstra and R. Tahri, "Homeomorphism groups and the topologist's sine curve," *Bulletin of the Polish Academy of Sciences. Mathematics*, vol. 58, no. 3, pp. 269–272, 2010.

[41] W. Maleszewski, "The arithmetic of the topologist's sine curve in cryptographic systems dedicated to iot devices," *TASK Quarterly. Scientific Bulletin of Academic Computer Centre in Gdansk*, vol. 23, no. 1, pp. 29–47, 2019.

[42] R. B. Lee, Z. Shi, Y. L. Yin, R. L. Rivest, and M. J. Robshaw, "On permutation operations in cipher design," in *International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004.*, vol. 2, pp. 569–577, IEEE, 2004.