Turing Machine Development for High-Secure Data Link Encoding in the Internet of Things Channel

Victor Tikhonov[1], Abdullah Taher[2], Serhii Tikhonov[3], Kateryna Shulakova[1,4], Vlad Hluschenko[1] and Andrii Chaika[1]

[1]*Department of Computer Engineering and Information Systems, State University of Intelligent Technologies and Telecommunications, Kuznechna Str. 1, 65023 Odesa, Ukraine*

[2]*Department of Electronic and Communication Engineering, College of Engineering, Al-Qadisiyah University, 58002 Qadisiyah, Iraq*

[3]*Department of Computer Systems, Odesa Polytechnic National University, Shevchenko Avenue 1, 65044 Odesa, Ukraine*

[4]*Anhalt University of Applied Sciences, Bernburger Str. 57, 06366 Köthen, Germany*

victor.tykhonov@suitt.edu.ua, abdallahqays@gmail.com, od.sergii.tihonov@gmail.com, katejojo29@gmail.com, vlad.gluschenko.97@gmail.com, chayka.and@gmail.com

Keywords: Internet of Things, Information Security, Data Link Encryption, Data Framing, Formal Grammar, Turing Machine, Packet-Based Streaming.

Abstract: The work considers the issues of data stream encoding in the IoT systems and networks in the context of information security provision, with particular focus on the data link layer in wireless and wired telecommunication channels. The problem state of the art shows a great progress in the sphere of IT cybersecurity based on Advanced Encryption Standard. However, new tasks and related threats emerge, such as unmanned mobile devices of mass disposable with remote control, for which known approaches are not reliable and efficient enough. In this respect, the common algorithms of data link frame encoding are studied, and original formalization of typical frames proposed for static and dynamic representation. An original method introduced for high-secure data encoding with variable frame structure. A formal grammar of an abstract Turing machine (TM) is developed for data link encoding, which is based on ternary line-signaling and three-bit command system. Constructed typical frame-patterns with the use of TM syntax. General principles are formulated for the high-secure frame encoding with variable structure for packet-based streaming on the data link layer with the use of TM algorithm. The results of the work intend to improve the mobile objects cyber-threats protection, as well as to remote vehicle control and other IoT real-time applications.

1 INTRODUCTION

The term "Internet of Things" (IoT) was first used in 1999 for physical objects with sensors connected to a packet network. Currently, IoT means the transition to a qualitatively new stage of the Internet evolution, which services extend to many devices, sensors and actuators [1].

IoT sensor networks are close to the mass user of Internet services, and this will dominate the Internet access networks. The further development of the Internet in the segments of concentration and distribution of traffic is formulated in the ITU concept of Next Generation Networks (NGN), which integrates a variety of data types and network services based on IP [2].

The core of the Internet is a transport system of optical lines based on high-speed coherent optical communications. Global satellite mobile communications networks such as Starlink are also being actively implemented. Each of these aspects has its own characteristics and challenges [3].

This paper considers information encoding at Data Link Layer (DLL) of IoT computer systems interfaces, for which a big challenge is personal data protection. Vulnerability links of IoT systems are radio channels, controller's software, traffic hubs. IoT cyber protection is complicated by a large amount of data and documents of various types, and each of them is a potential source of threats. Files "exe" and "com" may contain virus program bookmarks, which in themselves are not virus carriers, but refer to Web resources that contain them [4].

The variety of threats and countermeasures, as well as the growing risks of cyber-attacks in the modern world, require further theoretical and practical researches on the cyber protection methods of the IoT telecommunication channels, considering the characteristics of network interfaces at different levels of computer system architecture.

The Internet of Things cyber security architecture implies, that wireless connectivity is integrated into IoT devices and sensors, enabling them to transmit data to the network. The wired Ethernet along with the wireless WiFi links provide a standardized interface for these devices, allowing them to communicate with gateways or directly with cloud platforms [5].

Up to days, a ground-breaking progress achieved in network security technologies based on Advanced Encryption Standard (AES). However, new problems have arisen, such as unmanned mobile objects, for which the known approaches are not sufficient enough. Therefore, new researches needed in the IoT-cybersecurity.

The transmission of frames in a telecommunication channel can be taken as a sequence of commands and data for some abstract automaton that operates according to a certain algorithm and converts the incoming stream code into the structural-parametric code of individual frames.

The common principles of device operation, that transform information according to a certain algorithm, are studied by the automata theory. An important section of automata theory is the TM as a mathematical model that reduces the logical structure of an arbitrary processor to its basics (i.e. formalized rules for building grammatical structures on a set of alphabet symbols) [6].

In connection with the above, an actual scientific-applied task arises in the IoT field, that is, development the methods and algorithms for secure data transmission in the local networks and communication channels. This work intends to develop a formal grammar for the high-secure data encoding in the IoT data link channels on the base of an abstract TM algorithm.

Section 2 analyses the IoT cybersecurity state of the art. Section 3 formulates the objectives of the work. Section 4 presents formalization of the data link framework. Section 5 develops a TM grammar and general principles of high-secure packet-data encoding based on the ternary line-signaling and three-bit TM-command system. Section 6 summarizes results of the work.

2 THE IOT CYBER SECURITY STATE OF THE ART

Methods for transmitting, processing and protecting data in telecommunication systems and networks depend on the selected mathematical model for representing digital information. Let $A=\{a_n\}$, $n\in[1, N]$ be a set of discrete alphabet symbols used for interaction between network participants; $p(n, k)$, $(n,k)\in[1, N]$ - a priori probability of erroneously perceiving the symbol a_n as if it were a_k; $Q=\{p(n, k)\}$ - quantum uncertainty matrix.

The representation of digital information using a quantum uncertainty matrix Q corresponds to the quantum data model (QDM) in promising digital IT technologies that has been actively developing in recent years [7].

In the case of $p(n, k)\rightarrow 0$ when $n \neq k$, the matrix Q can be approximated by a diagonal matrix $Q\approx diag(P)$, where $P=\{p(n)\}$ is a quantum credibility vector for symbols, $p(n)$ is a priori probability of correct perception of the symbol a_n during the interaction of the parties. This case of representation the digital information using a quantum credibility vector P corresponds to a reduced quantum data model (RDM).

Finally, in the case of $p(n, k)\rightarrow 0$ for $n\neq k$ and $p(n,k)\rightarrow 1$ for $n=k$, the RDM-model turns into a classical data model (CDM). The difference between these three models (QDM, RDM and CDM) is rather conditional and not clearly defined. Commonly known IT-technologies primarily use the classical CDM-model of digital information representation.

We will consider here the cybersecurity issues with respect to the classical CDM-model of digital information representation, in which data reception-transmission errors are possible, but not critical.

Cybersecurity is a broadly used term, whose definitions are highly variable and often subjective. It includes organization, resources, processes, and structures used to protect cyberspace systems from occurrences that misalign property rights [8]. The cybersecurity also refers to improving the integrity of information management systems or infrastructure and addressing present and emerging challenges [9].

In this work, we will understand the cybersecurity for IoT networking as “a set of technologies, processes and practices to protect and defend networks, devices, software and data from attack, damage or unauthorized access” (Laboratory for Open Systems and Networks, Jožef Stefan Institute, Ljubljana [10].

The cybersecurity is becoming complex because of the exponential growth of interconnected devices, systems and networks. This is exacerbated by advances in the digital economy and infrastructure, leading to a significant growth of cyberattacks with serious consequences. In addition, researchers report the continued evolution of nation-state-affiliated and criminal adversaries, as well as the increasing sophistication of cyberattacks, which are finding new and invasive ways to target even the savviest of targets. This evolution is driving an increase in the number, scale and impact of cyberattacks, and necessitating the implementation of intelligence-driven cybersecurity to provide a dynamic defense against evolving cyberattacks and to manage big data. Advisory organizations, such as the National Institute of Standards and Technologies (NIST), are also encouraging the use of more proactive and adaptive approaches by shifting towards real-time assessments, continuous monitoring and data-driven analysis to identify, protect against, detect, respond to, and catalogue cyberattacks to prevent future security incidents [10].

One of the Internet protocol architects D. Clark pointed out cybersecurity as a serious problem of the global network. The Internet lacks of built-in security, and its shortcomings have resulted in a decreased ability to accommodate new technologies. We might just be at the point where the utility of the Internet stalls – and perhaps turns downward. It's time to rethink the Internet's basic architecture and start over with a fresh design. This is not about building a technology innovation but about architecture – pulling the pieces together in a different way. Improving the Internet is not so much about delivering the latest cool application, it's about survival. L. Peterson, a computer scientist at Princeton University, thinks that we've been on a track of incrementally making improvements to the Internet and fixing problems that we see. We see vulnerability, we try to patch it. That approach is one that has worked for 30 years. But there is reason to be concerned. Without a long-term plan, if you are just patching the next problem you see, you end up with an increasingly complex and brittle system. It makes new services difficult to employ. It makes it much harder to manage because of the added complexity of all these point solutions that have been added. T. Leighton (member of the President's Information Technology Advisory Committee) believes herewith, that there are more and more holes, and more resources are going to plugging the holes, and there are less resources being devoted to fundamentally changing the game, to changing the Internet [11].

During the global Network development, authoritative institutions proposed various models of Internet architecture. Each of these models reflects a specific vision of the organization on the current problems of the state and prospects of the Network, and from this point of view, defines multi-level interfaces for the open systems interconnection. Each layer and type of Internet model has its own cybersecurity vulnerabilities [12].

Commonly known are the following Internet models: 7-layer OSI/ISO reference model (1978-80), 4-layer TCP/IP model by IETF (1978-80), 3-layer NGN/ITU-T model (2004), 4-layer NGN/ITU-T model (2012), 7-layer IoT model by IoT World Forum (2014), 4-layer NIST-SP model (2016), 3-layer Industrial IoT model by IIC (2022).

Among the seven listed above Internet architecture models, three latter have been produced by new emerged institutions, and they reflect transition of traditional Internet to the Internet of Things. Similar to early Internet model's evolution by shifting the number of layers in the "7-4-3" pattern, the latter IoT-models reproduce this pattern but not in all. The last two models are of particular interest.

The NIST-SP model issued in 2016 by the Information Technology Laboratory (ITL) at the US National Institute of Standards and Technology (NIST) as a Special Publication 800-183 with particular focus on distributed computer systems security. The document offers "an underlying and foundational science to IoT based on a belief that IoT involves sensing, computing, communication, and actuation" [13].

Unlike previous Internet models, the NIST-model includes the human factor as a critical element of the IoT system security; it formulates five system primitives of the generalized Network of Things (NoT): Sensors, Aggregators, Communications, external Utilities (e-Utilities), Decision Triggers.

The first four ones refer to technical entities of the model, whereas Decision Trigger refers to human factor, which acts similarly to aggregator a special case of it. The model does not specify what is or is not a 'thing. In physical space, things consider humans, vehicles, residences, computer, switches, routers, smart devices, road networks, office buildings, etc. In virtual space, they consider software, social media threads, files, data streams, virtual machines, virtual networks, etc. An important primitive of the model is communication channel is a medium by which data is transmitted (wireless, wired, verbal, etc.). Since data is the "blood" of a NoT, communications are "veins" and "arteries" [13].

The latest known model of the Industrial Internet of Things (I-IoT), proposed by the I-IoE consortium in 2022, is quite consistent with the NIST model, and in a certain sense develops it. It contains three hierarchical categories: End Devices (sensors and actuators), I-IoT platform and Users (devices, individuals); two primitives of the NIST-model (Aggregators and Communications) merged into the "I-IoT platform".

Both IoT latest models consider communication channels as a critical element for the overall system security chain. There is a wide variety of telecommunication channels. In this paper we will focus on packet-based digital communication channels that operate in non-stationary or extreme conditions. This primarily concerns wireless channels for communication with mobile devices and objects, e.g. in WiFi networks.

Information in a digital packet network is supplied via a communication channel in the form of sequences of symbols (letters of the alphabet and syntactic signs), similar to grammatical text in school. Letters are combined into separate words according to spelling rules. Words are combined into sentences according to syntactic rules. A sequence of sentences forms a message (for example, a section of a document). An ordered set of messages forms a document file [14].

In a packet telecommunications channel, the role of grammatical sentences is played by frames, e.g. wired Ethernet or wireless WiFi frames, that operate on the physical and data link OSI-layers. The OSI network layer deals with IP-packets which carry distinct messages. The four upper OSI-layers are responsible for the file-documents delivering.

We will highlight three main components of the data protection process in the communication channel, regardless of the level of packet data transmission during remote interaction between the parties. These are:

a) method of plane text encrypting;
b) method of authentication and key provision;
c) protocol for packet generation.

Next, we will focus the IoT cybersecurity state of the art by tracking the WiFi wireless technology evolution, which is the most popular today in the Internet of Things applications. Over the past three decades, WiFi technology has evolved from a simple WEP-40 interface (Wired Equivalent Privacy, 1997, still widely used today), up to the modern WPA3 interface (WiFi Protected Access 3, certified since 2018), which is considered now the most advanced, and is only beginning to be deployed.

There are four main WiFi encrypted security protocols currently valid and available on the telecom market: WEP (IEEE 802.11 standard released in 1997 and ratified by WiFi Alliance in 1999); WPA (2003, IEEE 802.11i – an amendment to the original 802.11 standard), WPA2 (2004) and WPA3 (has been in use since 2004, WiFi Alliance certifying began since 2018). The features and characteristics of these interfaces are discussed below:

A) WEP interface:

1) *Method of plane text encrypting*. WEP is based on RC4 stream cipher (Rivest Cipher 4, designed in 1987 by R. Rivest from the American computer and network security company RSA) for data encryption with 8÷2048 bit key. Actually, WEP supports the RC4-encryption with 64/128-bit keys (40/104-bit static secret key-part + 24-bit initialization vector as a variative open key-part). Due to its speed and simplicity, RC4 is now widely used in Secure Socket Layer (SSL) and Transport Security Layer (TSL) of the IEEE 802.11 wireless LAN. However, because of its vulnerabilities, RC4 is no longer recommended for critical applications [15].
RC4 has 4 variants as follows.
   - SPRITZ - building the cryptographic hash functions and deterministic random bit generator.
   - RC4A - proposed to be faster and stronger than the average RC4 cipher, though it was found to have not truly random numbers in cipher.
   - VMPC - Variably Modified Permutation Composition – found to have not truly random numbers, like RC4A.
   - RC4A+ - an advanced version of RC4A that is longer and more complex than RC4 and RC4A, but is stronger as a result of its complexity as well.

   RC4 operates on a stream of data byte by byte with 64/128/256-bit keys. Yet, a flaw was found in RC4 where the 128-bit encryption key could be cracked in seconds. Another RC4-vulnerability was discovered in 2013 while it was being used as a workaround for a cipher block chaining issue (2011), while RC4 did not use itself the block-chain operational mode. It was found a way around RC4, with only a slight increase in processing power necessary in the previous RC4 attack. Now RC4 is considered a legacy method.

2) *Method of authentication and key provision.* The WEP-encryption for any local WiFi connection implies predefined static secret 40/104-bit key along with the 24-bit dynamic open key (initialization vector IV), that is (pseudo) randomly generated for each frame.
The WEP secret key can be provisioned automatically by the WiFi access point for a WiFi LAN host by arbitrary chose among the set of default keys when host authorization. Up to four default keys can be pre-set in the WiFi access point; default key is identified by their number 0, 1, 2, or 3 in the WEP-frame extended header. The WEP secret key can also be assigned as the WiFi LAN access password of 5/13 ASCII characters or 10/26 HEX characters (40/104 bits).

3) *WEP protocol for packet generation.* WEP was introduced as part of the original 802.11 standard released be IEEE in 1997, and ratified by WiFi Alliance in 1999 [16]. It specifies the set of medium access control (MAC) and physical layer (PHY) protocols for implementing wireless local area network (WLAN) communication. The WLAN MAC frame structure consist of a MAC header, frame body, and a frame check sequence (FCS). The 802.11 MAC frames can be either protocol version 0 or version 1. The algorithm of WEP-frame design is described in [17]. It is applicable for any frame or a packet that consist of a header and a payload:
Input: Header, Payload;
ICV= CRC(Header, Payload);
Data = (Payload, ICV);
IV = Initiation_Vector generator;
{**Key_Nr, Secret_key**} = Key_looup generator;
WEP_seed = (IV, Secret_key) // RC4 key;
Payload_Script = RC4(Data, WEP_seed);
WEP_Frame = (Header, IV, Key_Nr, Payload, ICV).
Output: WEP_Frame.

B) **WPA-interface.** The WPA protocol (WiFi Protected Access) is similar to WEP, but it is more secure due to increased *256-bit encryption* key. Besides, it uses an enhanced RC4-based TKIP cipher (TKIP) that includes three new features: a) 64-bit *message integrity check* (MIC) instead of CRC to re-initialize the sequence number each time when a new temporal key is used; b) *packet sequencing control* to protect against replay attacks (packets received out of order are rejected by AP; c) *per-packet key mixing* function that combines the secret key with the initialization vector before passing it to the cipher.

WEP, in comparison, merely concatenated the initialization vector to the root key, and passed this value to the RC4 routine. Compare to WEP authentication (Open system/Shared key), WPA-personal provides a *pre-shared key* (PSK) for Small Office Home Office (SOHO) wireless LAN.

C) **WPA2-interface.** WPA2 is a security standard for wireless networks based on the AES – Advanced Encryption Standard technology (NIST, 2001). It is used with the IEEE 802.11a/b/g/n/ac WiFi standards as the successor to WPA and has superseded WEP encryption [18]. Compare to WPA with the TKIP cipher, the WPA2 standard implemented an advanced cipher CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) that uses the AES 256-bit encryption algorithm; latter is one of the most robust encryption methods that is commercially available today.

While it is theoretically true that AES 256-bit encryption is harder to crack than AES 128-bit encryption, AES 128-bit encryption seems not to been cracked yet. Starting in August 2023, AES256-bit in cipher block chaining mode (AES256-CBC) will be the default encryption mode across all applications using Microsoft Purview Information. AES-256 encryption is virtually uncrackable using any brute-force method. It would take millions of years to break it using the current computing technology and capabilities [19].

D) **WPA3-interface.** The WPA3 standard was announced in 2018 by WiFi Alliance as a replacement to WPA2; it is now considered a cutting-edge security protocol to the market, and is a mandatory certification for WiFi CERTIFIED™ devices. WPA3 brings better protections to individual users by providing more robust password-based authentication, even when users choose passwords that fall short of typical complexity recommendations. This capability is enabled through Simultaneous Authentication of Equals (SAE, IEEE 802.11s standard issued in 2011 and superseded in 2012 when became part of the IEEE 802.11 standard of 2012). The technology is resistant to offline dictionary attacks where an adversary attempts to determine a network password by trying possible passwords without further network interaction.

The new standard uses 128-bit encryption in WPA3-Personal mode (192-bit in WPA3-Enterprise). The WPA3 standard also replaces the pre-shared key (PSK) with SAE as defined in IEEE 802.11-2016 resulting in a more secure initial key-exchange in personal mode. The WiFi Alliance also claims that WPA3 will mitigate security issues posed by weak passwords and simplify the setting up devices with no display interface. WPA3-Enterprise also offers Extensible Authentication Protocol for transport layer security (EAP-TLS) using Elliptic Curve Diffie-Hellman (ECDH) exchange and Elliptic Curve Digital Signature Algorithm (ECDSA) with 384-bit elliptic curve.

At the same time, the widespread adoption of WPA3 is a gradual process with potential limitations and challenges. Among them are weak backward compatibility and limited device support, vulnerabilities in implementation (errors or weaknesses in the deployment and configuration of WPA3 can undermine its security benefits). Regular firmware updates and adherence to security best practices are crucial to mitigate these potential weaknesses. However, when WPA3 gains wider adoption and attention from potential attackers, new vulnerabilities may emerge. The continuous discovery of security flaws and the need for timely patches and updates are ongoing challenges in maintaining the security of any wireless protocol, including WPA3. The weaknesses mentioned above do not undermine the overall benefits that WPA3 brings to wireless security [20].

The IoT-cybersecurity state of the art may conclude the following.

1) To date, principal results have been achieved in the field of security of networks and telecommunication channels. The top AES-256 encryption method is used to protect digital data, which is virtually uncrackable. The wireless channels of networks and systems commonly use a set of WiFi interfaces of different complexity and resistance to attacks like WEP, WPA, WPA2, while the most advanced WPA3 interface is now being actively implemented. At the same time, new threats and challenges emerge. One of them is provoked by an open packet structure for any WiFi standard; other things being equal, this factor facilitates to hack information at the data link layer.
2) Despite great achievements in the field of cybersecurity, we are now experiencing a major crisis in the global security system.
3) New operational and tactical tasks have arisen, for the solution of which conventional IT technologies turned out to be insufficiently effective. One of these unsolved problems is the creation of a simple and reliable interface for interaction with unmanned mobile objects of disposable use.

Based on the analysis of the surveyed publications, the objectives of this work are formulated in the context of protecting information on the data link layer in distributed systems of the IoT with the use of deterministic digital data model.

3 OBJECTIVES OF THE WORK

This work aims to develop a formal grammar for high-secure data link encoding with variable framework in the Internet of Things channel.

To achieve this, the following objectives are set:

1) Formalization the static and dynamic data link framework.
2) Construction a TM formal grammar for data structuring.
3) Developing a TM for high-secure data link encoding with variable framework.

4 FORMALIZATION THE DATA LINK FRAMEWORK

The wired and wireless LAN-technologies are widely used on the Data Link Layer (DLL) in today's IoT. As it was spoken above, the algorithms of DLL encryption and framing are tolerant to concrete frame structure, whenever the static frame consist of a header and a payload body. The WiFi wireless frame structure differs from the 802.3 wired standard, and it is more complicated. Therefore, we will further discuss the principal issues of DLL framing that refer to 802.3 wired LAN. The specifications DIXv2.0 (aka Ethernet-2) and IEEE-802.3 (aka Ethernet-3) historically remain two independent basic standards, that are not directly compatible. The later specification IEEE-802.3x unites both these two standards on the principle of backward compatibility. It defines the structure and parameters of the Data Link frame, invariant to the method of its transmission (structural-parametric code), as well as its mapping into a sequence of digital symbols for transmission over the communication channel – the stream code of the frame.

The Data Link streaming code is formed due to the frame structure and parameters, in accordance with related cable type and line-signaling. At the physical layer binary linear codes are popular (with two-digit physical signals, aka MII-codes) or three-digit (MLT-3, 100 BASE-TX). A simple binary code of type MII is the interference-resistant NRZ-PAM2 with a small frequency band; it is popular in the SerDes communications, which convert parallel code into serial one and vice versa. Its drawback is a DC component and weak synchronization in monotonous symbol sequences. The 10Mbitps interfaces implement a binary self-synchronized inverse Manchester code (MC-code of the IEEE 802.3 convention), in which any bit is at the same time a clock signal; the original MC-code is referred to as Thomas convention encoding. The interfaces 100BASE-T4/T1/1000BASE-T1 use the 8B6T block code in the 3-character PAM3 alphabet. For instance, the 100Base-T4 specification (IEEE 802.3u standard) uses the4-twisted pair cable of category 3; two of the four pairs are oriented in opposite directions, and the other two are switched in the direction of transmission. At any moment, three of the four pairs are transmitting data, and one is listening to the line for collisions. For every 2 cycles, three twisted pairs transmit a block of 6 ternary symbols (3^6 =729 numbers), which encode 256 octet-numbers according to redundant table scheme. The 1000BASE-T interface implements the PAM5 code, in which 2 positive and 2 negative amplitude levels encode 2 bits of information in a cycle (4-character alphabet), and zero-signal is not used. The 2.5GBASE-T interface uses the PAM16 code.

The DLL stream contains the structure-parametric components, embedded into a digital symbols sequence due to a certain formal grammar, considering the properties of the communication line and the counteracting parties' protocols. It can be shaped in a permanent commands sequence for an abstract digital processor. Let introduce the static Frame Parameter Code (**FPC)** as a system of vectors:

$$FPC = \{[AD], [ET], [D]\}, \quad (1)$$

where [AD] = (DMAC, SMAC) – vector of physical destination and source addresses (DMAC, SMAC) with 2×6=12 bytes in the network adaptor RAM; [ET] – vector of the DLL-frame type of 2, 5, 6 or 10 bytes in RAM, depending on the first two bytes, and besides, on the fifth byte of this vector; [D] = (DL, DAP) – vector of the payload data of the total size DL bytes in RAM, allocated by the Data Access Point (DAP).

The static **FPC** code is a temporary data record, which is necessary to generate a dynamic Frame Streaming Code (**FSC**) for the data transmission over the physical communication line between any two adjacent nodes within a LAN. The **FPC** code plus CRC-checksum (4 bytes) in the stream code of the 802.3x frame, must be 64÷1518 bytes length, (RFC 894, 1984). Hence,

$$|\mathbf{FPC}|+4 \le 1518, \text{ or } (2 \div 6+|ET|+DL)+4 \le 1518.$$

From this follows the data field length bound DL≤1502–|ET|. When |ET|=2, we have 0≤DL≤1500; at |ET|=10 we get (0≤DL≤1492) bytes.

Consider the three following **FPC** formats depending on the size of the ET field.

1) |ET|=2 bytes; this corresponds to the DIXv2.0 (Ethernet-2) frame, and the values ET=1536÷65534 encode the frame type.
2) |ET|=5 or 6 bytes; this corresponds to an IEEE-802.3 (Ethernet-3) frame, where the ET field is expanded into a D data field by 3 or 4 bytes of the LLC header (IEEE 802.2 specification). The first two bytes in the ET field determine the number of bytes of useful data DL=0÷1497. In the streaming code, the data field DL of less than (64–18) =46 bytes, has to be padded up to the minimal size of 46 bytes. The two least significant bits "xx" in the control byte of the extended ET field of the IEEE-802.3 frame define one of the three following modes of the LLC protocol:
   - **U** (Unnumbered frame, "xx"="11") – datagram mode of transmission of frames as independent packets of the channel level, LLC=3 bytes, |ET|=2+3=5 bytes;
   - **I** (Information frame, "xx"="00" or "10") – transmission of frames by multiplex streams over logical connections with LLN numbers (Logical Link Numbers) in the sixth byte of the extended ET field; LLC=4 bytes, |ET|=2+4=6 bytes;
   - **S** (Supervisor frame, "xx"="01") – frames of control messages contained in the sixth byte of the extended ET field; LLC=4 bytes, |ET|=6 bytes.
3) |ET|=10 bytes; this indicates the SNAP frame of the IEEE-802.3x standard (Sub-Network Access Protocol) with DL=0÷(1500–8) =1492 payload data size in the first 2 bytes of extended ET-header. Though, the DL-size in the stream code of such a frame must be at least (46–8) =38 bytes, otherwise it is padded to 38 bytes.

Unlike the IEEE-802.3 frame, the first 3 bytes of the LLC header in the extended ET-field of the SNAP standard frame contain a dummy LLC with a fixed value $(AA`AA`03)_{16}$ or $(AB\ `AB`03)_{16}$. The first 3 of the next 5 bytes of the SNAP header contain a 24-bit OUI (IEEE Organizationally Unique Identifier) that identifies the manufacturer and vendor of the network equipment. The last 2 bytes contain the 16-bit frame type number (similar to the ET field in the DIXv2.0 frame header). The expansion of the LLC header has been due to the gradual increase in frame types, despite the limited 6-bit type code in the 3rd byte of the previous LLC version. Since the hexadecimal values $(AA`AA)_{16}$ and $(AB`AB)_{16}$ for the first two bytes of the LLC were previously idle (reserved), they were used as "commands" for the extension of the LLC header.

The static DLL frame structure **FPC** (Frame Parameter Code) introduced above reflects the necessary and sufficient set of parameters for the node's interaction within a local network. An **FPC** code of a frame is formed at the OSI data link layer for temporal storage in the network adapter RAM for their further transfer over the physical layer link, or vice versa, to be passed to the network layer driver.

The **FPC** code cannot be directly transmitted into the channel under various technical circumstances. Instead, it must be injected into a digital symbol sequence of the Frame Streaming Code (**FSC**). The latter includes additional fields: an inter-frame guard interval (IFG), synchronization preamble (PRB), Start Frame Delimiter byte (SFD), and the Cyclic Redundancy Checksum (CRC, 4 bytes) from the vectors AD, ET and D of the **FPC**-code.

The **FSC**-code is limited in minimal size in order to distinguish the frame from the channel garbage when frames collision occurs. For that reason, the net-size data vector D of the **FPC** code is mapped into extendable gross-size payload data field PL of the **FSC**, while the set of parameters (AD, ET, PL, CRC) is kept for 64 bytes length or more (RFC 894). Thus, the Frame Streaming Code **FSC** (without the IFG guard interval) must be not less than 72 bytes.

Let define a *dynamic Data Link frame structure* **FSC** (*Frame Streaming Code*) as vector-system with a minimal size of 72 bytes:

$$\mathbf{FSC} = \{[PRB]_7, [SFD]_1, [\mathbf{FPC}]_{60}, [CRC]_4\}, \quad (2)$$

where PRB is preamble (7 bytes), SFD is the Start Frame Delimiter (1 byte), CRC is the checksum (4 bytes), and **FPC** is the spoken above static frame structure (Frame Parameter Code) of minimal 60 bytes length: $\mathbf{FPC}=\{[AD]_{12}, ([ET], [D])_{48}\}$.

5 TURING MACHINE FOR DATA LINK ENCODING WITH VARIABLE FRAMEWORK

The TM is commonly understood as an abstract digital processor, that is simple, intuitive, generic and formalizes computation performed by a human mind.

Most concrete digital processors perform the so called "program code", or just a "program" (consequence of commands and data allocated at ROM/RAM memory with the 8-bit structure of bytes); here, one byte is the elementary syntax unit (symbol) in program text. Instead, an abstract TM-processor can handle the program texts with symbols of 1 bit and more.

Generic TM operates with the ternary alphabet: an abstract "space" and two "letters" (0 and 1). Each ternary symbol carries the $\log_2(3) \approx 1.585$-bit information. While constructing the TM formal grammar for the DLL-interface, we will use the two formalized above objects: **FPC** (1) and **FSC** (2).

The first formal object (**FPC**) is the TM-basis of the high-secure protocols generation for data transmission within an arbitrary local area network (LAN). The DLL-interaction presumes the common 6-byte MAC-addressing, with no care of the DLL-standards limitations in the streaming code (e.g. the minimal frame length, fixed **FSC** shape, mandatory guard interval IFG, CRC format etc.).

The second formal object (**FSC**) is the TM-prototype for simulation the conventional data transmission and receiving via the spoken above 100BASE-T4 standard with ternary line-signalling PAM3 and 8B6T block encoding. The modern physical channels implement ternary symbol encoding also by the other modulation methods (e.g. 3-PSK in coherent communications).

5.1 The TM-Formal Grammar Definition

Consider the TM grammar (**TMG**) as (3):

$$\mathbf{TMG} = \{\mathbf{A}, \mathbf{Syn}, \mathbf{Sem}\}, \quad (3)$$

where $\mathbf{A}\{-, 0, 1\}$ is the TM ternary alphabet of abstract line-signalling symbols: syntax sign "space" (−), and two "letters" (0, 1); **Syn** is the formal TM-syntax; **Sem** is the unformal TM-semantic.

Next, a TM-code $\mathbf{TMC}\{\mathbf{S}\}$ is defined on the set **S** of ternary alphabet symbols $\mathbf{s}\in(\mathbf{A})$, along with the TM-grammar syntax:

1) An TM-word is a non-empty sequence of "letters" 0 or 1 separated by one or more "spaces".

2) An TM-command is a TM-word with no more than three "letters"; others words are TM-data.
3) The TM-command basic set is shown in Figure 1. The overall number of TM-commands equals 14, including two 1-letter words (0 and 1), four 2-letter words (00,01,10,11) and eight 3-letter words (000÷111). The first 13 command of the basic set are tags; the command 14 is reserved for extended commands (13 additional commands plus next ones).
4) The common punctuation tags follow the syntax rules of ordinary spoken languages. Two or more “spaces” are equivalent a single one; parenthesis (), brackets [] and braces {} must be paired; repetition of tags prohibited, etc.
5) Nesting of tags {{...}} or [[...]] prohibited.
6) Tags {...} define an opened/closed sequence of multiple framing objects FPC and/or FSC.
7) The **TMC** codes are consequently transmitted in sessions {...}:

$$\{...\}\ \{TMC_1, TMC_2 ... TMC_N\}\ \{...\}-,\ N\geq 0.$$

8) The **TMC** may include the components:

TMC= [AD(...); DAT(...); CRC(...)],

where AD is the address vector (DMAC, SMAC); DAT is data vector (Type, Payload); CRC is checksum of AD and DAT. All the **TMC** components could be dummy, as well as the **TMC** itself.

№	Code	Syntax	Command
1	0	.	Dot
2	1	,	Comma
3	00	;	Semicolon
4	01	:	Colon
5	10	(	Left Parenthesis
6	11	)	Right Parenthesis
7	000	AD	Address Vector
8	001	DAT	Data
9	010	CRC	Checksum
10	011	[	Left Bracket
11	100	]	Right Bracket
12	101	{	Left Brace
13	110	}	Right Brace
14	111	$	Extension

Figure 1: The basic set of TM commands.

In contrast to common grammars with clearly defined rules, the TM-semantic is beyond the syntax formalism. Instead, it is rather to be introduced by related patterns collection. In practice, the semantic of a formal grammar can be embodied in appropriate algorithms and related computer programs made by human mind, or due to machine learning techniques, based on the sufficient AI training sequence of human mind design. Further on, we bring several TMC patterns of data transmission in a digital channel:

A) Pattern 1. IoT Real-Time data Streaming.

The telemetry real-time data stream (RTS) is characteristic in a sequence of short messages (one or several bytes each one) and fast label-switching. The following pattern fits this.

TMC = ... {RTS.RTS.RTS} ...
RTS = [AD 0001 DAT xxxxxxxx CRC].

The telemetry stream is designed as an open sequence of sessions {...}; each session transmits three RTS segments (i.e. frame parameter codes). Each **FPC** includes the 4-bit label (0001) of the predefined logical connection, 8-bit data field, and 1-bit CRC (parity bit “p”). In case of data field “10101011”, the **FPC** TM ternary code is

–011–000–0001–001–10101011–010–1–100–

Here, 37 ternary symbols of the **FPC** line-code occupy $37\times\log_2(3) \approx 58.6$-bit, while carrying 8-bit payload with information efficiency 8/58.6 ≈0.137, compare to DIXv2.0 framing with 10 times worse efficiency of 8/672≈0.012:
12(IFG)+7(PRB)+1(FSD)+6(DMAC)+6(SMAC) + +2(ET)+46(PLD)+4(FCS) = 84bytes = 672 bit.

B) Pattern 2. DIXv2.0 Frame Simulation.

TMC = ... {**FSC**.**FSC**.**FSC**} ...
FSC= DMAC SMAC ET Data CRC.

A session { } is presented by 3 consequent frames, each of 5 semantic parameters in regular format. The Data field has 0.5÷1500 byte without padding. The extra **FSC**-fields (IFG, PRB, SFD) are not needed. According to above TM syntax, the data-words are of 4 and more bits.

C) Pattern 3. Frame Parameter Transmission.

TMC = {AD (DMAC, SMAC) DAT(Data)}.

A session { } is presented by 2 pseudo-frames: the first one carries DMAC, SMAC, and the second – 0.5÷1500 byte of Data field without padding. This pattern can be used in case of low signal-to-noise rate (SNR) in the channel.

5.2 Principles of TM-Based High-Secure Encoding

The high-secure data link encoding with TM-encryption technique includes the following.

1) The set of TM-codes allows variety of TM-based formal grammars for encoding the

DLL-streaming data by permutation the rows in the TM-codes table. So, a pre-shared permutation number can be used as an extra secret key for DLL-stream encryption, in addition to conventional encryption keys.

2) Specific unmanned disposable mobile objects with remote control may allow to provide a simple and low-cost but unique encryption scheme at the data link layer using the one-time secret keys generated and pre-shared before their launching.
3) Based on the DLL-frameworks, a family of untypical frames can be outworked and secure-indexed to provide each object-mission by a unique subset of frameworks and related secret keys. This makes the channel harder to hack.

6 CONCLUSIONS

The IoT-cybersecurity state of the art indicates significant progress in network security both in wired and wireless communications. Nevertheless, new tasks and security challenges emerge, for which known approaches are not sufficient enough.

The work introduces an original TM formal grammar for high-secure data link encoding to increase the wireless and wired channel protection in special IoT-applications.

General principles proposed for data link encoding by the use of TM-encryption with variable framework to enable an unpredictable dynamic switching of the streaming frames structure and make the communication channel harder to hack.

The concrete TM-based security mechanisms are the objects of further researches. The results of the work intend to apply in unmanned disposable mobile objects with remote control and special IoT-systems.

ACKNOWLEDGMENTS

We acknowledge support by the German Research Foundation (Deutsche Forschungsgemeinschaft DFG) - and the Open Access Publishing Fund of Anhalt University of Applied Sciences.

REFERENCES

[1] D. Hanes, et al., "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things," Cisco Press, USA, 2017, pp. 543.

[2] International Telecommunication Union, "ITU-T's Definition of NGN," 2004.

[3] S. Wali, "The Internet: Opportunities and Challenges," 2022.

[4] M. Aqeel, et al., "A Review of Security and Privacy Concerns in the Internet of Things (IoT)," 2022.

[5] IEEE Computer Society, "The Internet of Things (IoT) to Cloud: Computing and Communication Gateways," 2023.

[6] J.E. Hopcroft, et al., "Introduction to Automata Theory, Languages, and Computation," 3rd ed., Pearson Education, 2008, pp. 554, [Online]. Available: https://openlibrary.org/books/OL24357442M/Introduction_to_automata_theory_languages_and_computation.

[7] F.J. Duarte and T.S. Taylor, "Quantum entanglement in matrix notation," in "Quantum Entanglement Engineering and Applications," 2021, ch. 7, pp. 1-9.

[8] D. Craigen, et al., "Defining Cybersecurity," 2014, [Online]. Available: https://www.researchgate.com/publication/267631801_Defining_Cybersecurity.

[9] F. Schiliro, "Towards a Contemporary Definition of Cybersecurity," 2022, [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2302/2302.02274.pdf.

[10] R. Kaur, D. Gabrijelčič, and T. Klobučar, "Artificial intelligence for cybersecurity: Literature review and future research directions," Elsevier, Information Fusion, vol. 97, 2023.

[11] F.D. Talbot, "The Internet is broken," MIT Technology Review, 2006, [Online]. Available: https://www.technologyreview.com/2006/02/15/229667/the-internet-is-broken/.

[12] A.C. Franco da Silva and P. Hirmer, "Models for Internet of Things Environments. A Survey," Information, vol. 11, no. 10, p. 487, 2020.

[13] J. Voas, "NIST Special Publication 800-183. Networks of Things," Computer Security Division, Information Technology Laboratory, 2016.

[14] C. Vleugels, "Human communication," [Online]. Available: https://osuokc.edu/sites/default/files/documents/arts/SPCH-1113-Speech-%26-Communication.pdf.

[15] V. Kota, "RC4 Cipher," 2021, [Online]. Available: https://www.linkedin.com/pulse/rc4-cipher-venkata-siva-naga-sai-kota?trk=public_profile_article view.

[16] D. Ghimiray, "Wi-Fi Security: WEP vs WPA or WPA2," [Online]. Available: https://www.avast.com/c-wep-vs-wpa-or-wpa2.

[17] G. Singh, "WEP Encryption and Its Vulnerability in Detail," 2023, [Online]. Available: https://tbhaxor.com/wep-encryption-in-detail.

[18] "How WPA2 differs to WEP and WPA," [Online]. Available: https://www.nfon.com/en/get-started/cloud-telephony/lexicon/knowledge-base-detail/wpa2.

[19] IDERA Inc., "What is AES 256-bit Encryption?", 2024, [Online]. Available: https://www.idera.com/aes-256-bit-encryption.

[20] "What is WPA3 vs. WPA2?" [Online]. Available: https://www.portnox.com/cybersecurity-101/wpa3.