Proceedings of the International Conference on
Information Technologies (InfoTech-2017)
20-21 September 2017, Bulgaria

Invited paper

THE METHOD OF DATA INTEGRITY ASSURANCE FOR INCREASING IOT INFRASTRUCTURE SECURITY

Vladimir Pevnev, Yegor Novakov, Mikhail Tsuranov, Vyacheslav Kharchenko

Department of Computer Systems and Networks
National Aerospace University named after N.E. Zhukovsky "KhAI"
e-mails: v.kharchenko@csn.khai.edu, m.tsuranov@csn.khai.edu
Ukraine

Abstract: Cybersecurity problems of modern IoT devices are reviewed in the article. The data integrity problem of IoT infrastructure was also highlighted in the article. Effective data transmission speed in IoT networks research method are reviewed in the article. Error grouping factor according to existing networks analysis was introduced. Critical parameters of backbone data transmission networks are analyzed. Application of noise-immune codes for data integrity securing in IoT infrastructure energy efficiency was researched. Possibility of special code tables application for traffic that is being transmitted by IoT devices reducing and for noise-immunity increasing is presented.
Key words: noise-immune encoding, data integrity, energy efficiency, error grouping model, IoT.

1. INTRODUCTION

Information security specialists started to focus more on cyberattacks opposition recently. The reasons are both global use of computers and unauthorized access to cyber resources possibilities growth. It must be said, that in 2012 the computer systems of the agency in charge of America's nuclear weapons stockpile are "under constant attack" and face millions of hacking attempts daily, according to officials at the National Nuclear Security Administration [1]. Devices of “Smart home” technology and IoT are the most exposed to attacks. There will be more than 25 billion of the devices by 2020, according to Gartner [2], therefore these technologies attract attention of cybercriminals and large-scale companies. Symantec has been secured more than 1 billion IoT devices by the end of 2015, according to their data [3].

Three general elements of IoT are things, Internet and connection system. A lot of various data is transmitted through the system, therefore system efficiency and the main advantage – the ability to create self-developing and self-improving systems, which can blur the line between physical and digital worlds depend on transmitted data integrity [4].

Amazon developed several cloud technologies, allowing to ease teaching process and IoT devices use significantly to solve the problem of data transmission in IoT. Among the most important technologies, it is necessary to emphasize:

- AWS IoT – is an automated cloud platform that allows connected devices to interact with cloud applications and other devices easily and safe. AWS IoT supports billions of devices and trillions of messages, can process and route the messages to AWS endpoints and other devices reliably and safely [5];
- AWS Greengrass – is software solution for safe local computations management, messages transfer and data caching on connected devices [6].

Emphasized tools allow to manage the data transmission between devices and central. However, the data integrity mechanisms management falls on telecommunication services providers, who use traditional network protocols mechanisms, when using these technologies. This is the reason why a lot of authors research IoT data integrity problem. For example, Arlen Baker writes: When we think about data of an IoT device we should think about not only the data being generated or used by it but also its own programming data, this including all aspects of program software, configuration parameters and operating system software. To guide the process of integrity it is helpful to consider three different states that data can exist, namely in motion, at rest and in process [7]. The author of work [7] says: Any breach of data integrity will mean that an IoT device cannot operate correctly but it also potentially exposes the device to being exploited and become a compromised platform from which other attacks can be launched. However, Arlen Baker does not suggest supporting the integrity, he suggests controlling it using cryptographic facilities: The usual method of verifying the integrity of data is by a mathematical algorithm called a hash, of which the secure hash algorithms (SHA) is most popular [7].

Martin Ruubel attaches significant importance to the IoT data integrity in his article [8]: Strangely all the security focus seems to be on privacy, as if the public disclosure of the contents of your fridge is something to be feared. We argue that integrity is by far the more important component of the CIA (Confidentiality, Integrity, and Availability) security triad. Privacy might cost you some embarrassment but integrity (of your medical devices, of your car's braking system, of your flight's altimeter, of your power supply) can easily cost you your life [8]. Nevertheless, author suggests solving the data integrity problem along the traditional lines: PKI and KSI are both just tools for data security and have complementary roles. PKI is best used for authenticating people on a network and establishing secure communications; KSI is best used for acting as an integrity proof for data at rest, providing

a mathematical verifiable audit trail for what happens when without a trusted party or key in sight [8]. That allows us to control the integrity, not to secure it.

Sean Kanuck, national intelligence officer for cyber issues at the National Intelligence Council in the ODNI recently emphasized that integrity (of data, of processes, of infrastructure, of software) is the biggest threat in cyberspace [9].

Based on all the above, the integrity problem must be attached more importance. And it must be secured, not controlled. Using simple integrity control, message signature jamming may lead to full message rejection that will increase data processing from IoT devices time. That delay can be critical enough for some of them (life-support system, secondary breaking system).

It is needed to evaluate the most effective data transmission speed in packet networks since this kind of networks has gained the greatest propagating.

2. EFFECTIVE DATA TRANSMISSION SPEED IN PACKET NETWORKS

2.1. Effective speed evaluation

It can be used following definitions to evaluate effective data transmission speed in packet networks, provided all devices operate correctly [10]:

$$R_e = f(R_0, V_k, n_p, t_r, \varepsilon, P_e, z, K_p), \quad (1)$$

where R_0 - data transmission speed that has been proved theoretically; V_k - code speed; n_p - data packet length; t_r - signal propagation speed in relation to time for packet analyzing and negative acknowledgement; P_e - unit data element error probability; z - negative acknowledgements count, K_p.- binary value for noise-immune codes use during message transmission fact.

Take the case where packet has length n_p and contains k of informational elements ($V_k = k / n_p$), and P is packet error probability. Then average packet transmission time can be defined as below considering z possible negative acknowledgements:

$$t_l = T_p \sum_{i=1}^{z} P_i, \quad (2)$$

where T_p – non-recurring packet transmission time.

P^z - transmission bus rejection due to jamming probability in relation to time for restoring T_v. Considering these comments, effective data transmission speed can be defined as:

$$R_e = V_k n_p \left[T_p \sum_{i=1}^{z-1} P_i + p^z (T_p + T_v) \right]^{-1}. \quad (3)$$

The following expression is right if noise-immune codes (K_p=1) that can identify errors is used in transmission:

$$P \approx P(\geq 1, n_p) = P_e n_p^{\varepsilon}, \quad (4)$$

where $P(\geq 1, n_p)$ - is distortion of one and more elements in packet on length n_p possibility.

If we substitute expression (4) in expression (3) obtaining expression:

$$R_e = V_k R_o \frac{n_p (1 - P_e n_p^{\varepsilon})}{(R_o t_A + n_p) + R_o T_v (P_e n_p^{\varepsilon})^z} \quad . \quad (5)$$

where $1 - P_e n_p^{\varepsilon}$ - is rate of decreasing R_e due to noise effect;

Expression (5) demonstrates an effect of general factors, that effect on R_e decreasing. Multiplier $1 - P_e n_p^{\varepsilon}$ represents noise effect on R_e decreasing. Summand $R_o t_A + n_p$ in denominator represents loss amount of R_e due to time that receiver has spent on message analysis and time that sender was listened for confirmation of acceptance t_A. Second summand $R_o T_V (P_e n_p^{\varepsilon})^z$ specifies amount of losses R_e that are caused by possible noise effect and exceeding maximum permitted negative acknowledgements amount z.

Analysis of expression (5) shows that depending on packet length effective speed has maximum, which value depends on R_o, V_k, t_A, z, P_e, ε and T_V. Based on service profile, the R_e value defines real network devices throughput and packet transmission time, as well as the effect of noise on network. Parameter R_e binds load stress parameter to service quality. The maximum effective transmission speed R_{emax} can be achieved at some value of optimal packet length n_{popt}, and is defined from expression $dR_e / dn_p = 0$ that doesn't have analytical solution relatively to n_p even if hardware had high reliability. Effective information transmission speed R_e has (depending on n_p) maximum value R_{emax}, that reduces to n_{popt} values with the increase of R_o. Extremum of $R_e = f(n_p)$ dependence is the is sharper when R_o is greater. This reflects less criticality of n_p choosing when R_o decreases.

2.2. Errors grouping factor

Error grouping can be defined as error grouping rate ε or grouping factor [11] in communication. The factor is defined statistically for every type of data transmission networks. The grouping factor has value in range $0 \leq \varepsilon \leq 1$. Value of $\varepsilon = 0$ corresponds to independent errors distribution. Errors are grouped in packets, when grouping rate ε increased, and when the rate is maximum ($\varepsilon = 1$), errors are grouped in one packet. According to errors grouping rate, there are channels with low, middle and high grouping rate defined, where $\varepsilon \leq 0{,}3$; $0{,}3 < \varepsilon \leq 0.5$; $\varepsilon > 0.5$.

2.3. Messages transmission quality parameters in international network

According to [12] there are defined such BER parameters in international network as normal – BER<10^{-6}; low – 10^{-6} ≤BER< 10^{-3} (accident-sensitive state); unallowable – BER ≥ 10-3 (accident state).

It must be said that international standards (except BER) define other channels quality parameters [12]: Errored Second (ES) и Severely Errored Second (SES). Errored Second Ratio (ESR) and Severely Errored Second Ration (SESR) are also been used. ESR and SESR, are defined as ratio of number errored seconds and number of severely errored seconds to the total number of seconds in the measurement. ESR=0,02, and SESR=0,001 in modern networks [12]. Adaptive algorithm that was suggested in paragraph 2.1 can greatly effect on the channels quality parameters that was specified in paragraphs 2.2 and 2.3.

3. RESEARCH OF NOISE-IMMUNE CODES ENERGY EFFICIENCY IN IoT

The main objective of the research is getting sufficient results in estimating the energy efficiency of the noise-immune codes use in various implementations and communication channels.

To achieve the objective following tasks have been accomplished:

- existing estimating methods of noise-immune codes efficiency have been analyzed;
- complex index of estimating of noise-immune codes efficiency has been developed. The index includes energy efficiency estimating depending on use of codes in various communication channels and with various encoding and decoding devices.

3.1. Main point of estimating of code use in IoT energy efficiency technique

When the methods of estimating of noise-immune code use efficiency are developed, it’s necessary to use complex efficiency indexes that must consider the following statements:

- ability to design codes for any communication channel, not only behavior of codes in real system;
- ability to abstract from real hardware platform when using complex indexes;
- have ability to compare code speed both in software and hardware implementation;
- the energy efficiency index in complex index on every code estimating state.

Following complex index is suggested for modern noise-immune codes efficiency analysis:

$$K_к = K_{w1}\frac{K_1}{K_1+K_2} + K_{w2}\frac{F_{cpu}}{t_\Sigma c} + K_{w3}\frac{F_{cpu}}{t_\Sigma dc} + K_{w4}\frac{Ko_{sn}}{Ko_{tot}} + K_{w5}\frac{K_{er}}{K_{er}+K_{fer}} + K_{w6}\frac{K_{мах}}{K_{tot}} + K_{w7}\frac{P_{nr}}{P_{st}} + K_{w8}E_b(K_1+K_2) \quad (6)$$

where K_{w1}, K_{w2}, K_{w3}, K_{w4}, K_{w5}, K_{w6}, K_{w7}, K_{w8} – are weighting factors that are defined by experts;

K_1 – the number of informational symbols;

K_2 – the number of checking symbols;

F_{cpu} - CPU frequency that is measured in cycles;

$t_\Sigma c$ - total time, that is needed to complete encoding operations of noise-immune encoding algorithm with equal data packets, that is measured in CPU cycles;

$t_\Sigma dc$ - total time, that is needed to complete decoding operations of noise-immune encoding algorithm with equal data packets and communication channels, that is measured in CPU cycles;

Ko_{sn} – the number of operations in the algorithm, that are completing simultaneously;

Ko_{tot} – total number of operations in the algorithm;

K_{er} – error number, that have occurred as a result of transmission through communication channel;

K_{fer} – the number of errors, that have been proved theoretically and fixed;

K_{tot} - total number of bits for transmission using the algorithm;

$K_{мах}$ – maximum possible number of error bits in the algorithm that could be corrected;

P_{nr} – transmitter's capacity during use of the noise-immune code;

P_{st} – standard transmitter's capacity;

E_b – noise-immune code energy efficiency per bit of transmitted information.

The complex efficiency index consists of eight indexes. Let us look more closely at indexes that are responsible for energy efficiency:

a) $\frac{P_{nr}}{P_{st}}$ – practical index, that is formed based on transmitter's capacity in regular mode and with use of chosen noise-immune codes. The index shows energy gain rate when used on specific communication channel;

b) $\frac{E_b}{K_1+K_2}$ – practical index, that is formed based on energy cost of encoding and decoding of one bit of information per total number of transmitted data.

The novelty of obtained results is shown below:

a) the methodology of use of noise-immune codes in IoT communication channels efficiency and energy efficiency estimation, that is based on use of complex efficiency indexes, that combine precise mathematical efficiency and energy efficiency in IoT communication channels estimation abilities and methods of expert estimation, that correlate efficiency of noise-immune systems embedding results in various

communication channels. The most critical noise-immune codes' parameters in respect to energy efficiency were analyzed too;

b) the methodology of noise-immune encoding estimation according to results of mathematical errors modelling in communication channels gained further improvement;

3.2. Practical use of noise-immune codes efficiency estimation in IoT

Practical relevance of received results lies in the following:

- they let us raise estimation completeness of energy efficiency of various noise-immune codes embedding in IoT infrastructure;
- they are base for developing of informational security systems that have integrity control and can analyze the energy efficiency of its use.

4. CODE TABLES USE FOR DATA TRANSMISSION IN IOT INFRASTRUCTURE

The most important factor is encoding block size, when using encoding in IoT. In considering code tables, which are shown in [14], the most relevant encoding block size is 4 bits [15]. The main argument for choosing that size is concept of restoration of bits that where received by mistake – a large controlled block size results large variants must be scanned.

As a result of modelling [16] it was discovered, that the probability of existing controlled blocks with more than 1 mistake is not higher than 0.02, when transmitting 1 KB message. And the probability of mistake in one symbol being 10^{-3}. At the same time the average errored controlled block count was a little more than 8 and the maximum count was 18.

Based on experiment results [16], it appears that grouped errors become single if controlled blocks are small enough. From this point of view 6-bit code tables use is optimal in channels with error probability $P_{ош} \approx 10^{-3}$.

From the point of view of data structuring in computers we must divide the introduced bit sequence on two parts for 3 bits in each and add one parity bit. Therefore, a pattern, that includes controlled blocks size in 4 bits, may be built. Also 8-bit sequence is suitable for further processing and transmission.

4.1. The procedure of setting up the code tables

The procedure of setting up the code tables provided that 6 bits are used per symbol encoding plus 2 bits for noise-immune encoding is shown in [14, 15]. Let us analyze typical code table of provided method. Codes of translating to different alphabet are specified in table 1. They are used in method specified in [14]. Codes of

translating to different alphabet have the highest interest, because errored decoding that codes leads to irreversible losses and unavailability to decode whole message text. For example, when you send 111100 (translating to alphabet 4) the forth bit would be errored and we receive 111000 (translating to alphabet 3). Because of it, all following transmitted messages would be decoded incorrectly and repeated symbols transmission or repeated decoding of previously received messages would be needed because of verification control. The mistake could be solved by 2 control bits, but if the mistake will occur in two mini-blocks 4 bits each simultaneously it would be impossible to correct the message.

Table 1. Code symbols for different alphabet translation

Code table index	***Binary view***	***Alphabet symbols***
51	*110011*	*Translation to alphabet 1*
52	*110100*	*Translation to alphabet 2*
56	*111000*	*Translation to alphabet 3*
60	*111100*	*Translation to alphabet 4*
63	*111111*	*Translation to alphabet 5*

Method that is described in [14, 15] has other sufficient disadvantages except listed above:

- alphabet translation symbols are duplicated in all code tables, that reduces symbols number, that could be selected for other symbols significantly;
- limited number of notes for translation symbols to another alphabet can be separated, that reduces simultaneously used alphabets number;
- minimal code distance between alphabet translation symbols is 1, which does not let to discover single error. If checksum is used the minimal code distance will be increased to 2, but that would not let us to discover pair errors.

4.2. The procedure of setting up the translation table

To eliminate defects listed above the authors have suggested method consisting in forming separate code table with symbols of translation to different alphabet. The method consisting in creating special table, that contains codes of all alphabets that are used in system, instead of putting code of translation to another alphabet in every code table. In every alphabet exist single combination for translation to code table. The authors suggest using combination that are filled with zeros for translation code security increasing. Using one transition combination allows you to allocate additional space in tables that can be used for the most commonly used symbols or special commands of IoT devices. This also significantly reduces the probability of wrong decoding of the transition combination due to interference impact.

After receiving that combination decoder would know that following byte should be decoded from another code table. Code tables for 9 alphabets example is reviewed in table 2.

Table 2. Code table for translation to different alphabet symbols

Code table index	***Binary view***	***Alphabet symbols***
0	*100001*	*Translation to alphabet 1*
1	*001100*	*Translation to alphabet 2*
2	*110000*	*Translation to alphabet 3*
3	*110101*	*Translation to alphabet 4*
4	*111111*	*Translation to alphabet 5*
5	*011110*	*Translation to alphabet. 6*
6	*101101*	*Translation to alphabet 7*
7	*010010*	*Translation to alphabet 8*
8	*000011*	*Translation to alphabet. 9*

Minimal code distance for every combination without control bits is 2, as shown in table 2. That let us to discover single errors. In case of two control bits use, code distance increases to 4. That factor let us consider that using the suggested code table for translation to different alphabet symbols allows us to discover double error. This fact let us increase translation to different alphabet symbols decoding validity and decrease probability of repeated transmission because of wrong symbols decoding.

It should also be noted that 9 alphabets, that are reviewed in table 2, almost twice the number of alphabets that used in the method tables [14]. Using suggested encoding method let us increase the number of simultaneously used alphabets or symbols significantly. This let us use encoding procedure suggested in [14] in international data transmission systems and with some national alphabets more effectively.

Another advantage of suggested procedure in comparison with the one stated in [14] is less data redundancy in every code table as far as only one code combination, which is translation to alphabet table, is repeated in every table.

5. CONCLUSION

The presented complex efficiency index lets us to provide a comparative analysis of noise-immune codes that are used in different communication channels. Both results of mathematical modelling of communication channels and results of real codes use were taken into consideration in code quality evaluation. It is important that code energy efficiency is also considered in code embedding abilities evaluation. The index considers energy efficiency of code use both in the communication channel and during coding and encoding procedures.

Based on error models' experimental studies and speed of noise-immune codes, considering received data and using developed efficiency index it should be pointed out that presented noise-immune code meets requirements of green technologies and uses energy component as effectively as possible in most cases. Optimal table constructing let us to decrease the redundancy of transferred messages. New translation table inclusion let us to increase resistance to translating to different alphabet by mistake. This is achieved through code distance between translation table elements increasing.

Using the proposed methods will significantly increase the speed and efficiency of the transmission of control signals in the IoT infrastructure, and choose the most appropriate method for securing data integrity at the design stage.

REFERENCES

[1] Jason Koebler (**2012**) *U.S. Nukes Face Up to 10 Million Cyber Attacks Daily*, (*available at*: https://www.usnews.com/news/articles/2012/03/20/us-nukes-face-up-to-10-million-cyber-attacks-daily).

[2] Gartner (**2015**) *Gartner Says 4.9 Billion Connected "Things" Will Be in Use in 2015*, (*available at*: http://www.gartner.com/newsroom/id/2905717)

[3] Symantec (**2015**) *Symantec Secures More Than 1 Billion Internet of Things (IoT) Devices*, (*available at*: https://www.symantec.com/about/newsroom/press-releases/2015/symantec_0825_013).

[4] AWS (**2017**) *AWS IoT*, (*available at*: https://aws.amazon.com/ru/iot).

[5] Amazon IoT (**2017***) Amazon IoT-platform*, (*available at*: https://aws.amazon.com/ru/iot-platform).

[6] Greengrass (**2017**) *AWS greengrass*, (*available at*: https://aws.amazon.com/ru/greengrass).

[7] Baker Arlen (**2013**): *Maintaining data integrity in Internet of Things applications*, (*available at*: http://files.iccmedia.com/pdf/windriver160823.pdf).

[8] Ruubel Martin (**2013**) *Privacy and Integrity on the Internet of Things*, (*available at*: https://guardtime.com/blog/privacy-and-integrity-on-the-internet-of-things-if-all-you-have-is-a-pki-hammer-dot-dot-dot).

[9] Robert K. Ackerman (**2013**). *Blog: Data Integrity Is the Biggest Threat in Cyberspace*, (*available at*: http://www.afcea.org/content/?q=node/11438).

[10] Pevnev, V. Y., Tsuranov, M. V., Logvinenko, M. F. (**2016**). Noise-immune codes evaluation methodics, *Radio-electronic and computer systems: technical science, magazine*. vol. 5, pp. 165-170.

[11] Morozov, V. G., Purtov ,L. P., Zamriy, A. S. (**1981**). Experimental data generalization bases on possibility and error grouping index. *Communication devices technics*, edition 4(2), pp.53-60

[12] M.2100 (**2003**) *M.2100: Performance limits for bringing-into-service and maintenance of international multi-operator PDH paths and connections* (*available at*: https://www.itu.int/rec/T-REC-M.2100/en).

[13] Study report (**2015**) Science fundamentals, green computing and communication methods and facilities. Volume 1. Regulations of green computing and communication methods and facilities under resources limitation. Study report (government registration index 0115U000996). – *Kharkov: National Aerospace University named after N.E. Zhukovsky "KhAI"*, 246 p.

[14] Pevnev, V. Y., Yacenko, I. L. (**2002**). Aboute the method of text information.*Visnyk Zhitomir engineer-technical institute. Zhitomir.* vol. IV (23), pp.206-209.

[15] Pevnev, V. Y, Yacenko, I. L. (**2003**). Date restoring in data transfer in DCS. *Visnyk. Kremenchug Government Polythechnical Institute.* vol. 3/2003 (20), pp. 19-21.

[16] Pevnev, V. Y., Tsuranov, M. V. (**2011**). Grouped errors in communication channels models experimental researches. *Visnyk NTU "KhPI". Science works digest. Kharkiv: NTU "KhPI"*, vol. 49, pp. 115-121.