Security
of Infocommunication Systems
and Internet of Things

2026 Vol 4, No 1

https://doi.org/10.31861/sisiot2026.1.01020

Received 08 June 2026; revised 19 June 2026; accepted 26 June 2026; published 30 June 2026

Key Distribution in Wireless Sensor Networks Using Random Channel Selection

Serhii Nichyi[1,*], Petro Shpatar[2] and Bohdan Nichyi[2]

[1]Department of Electronics and Power Engineering, Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine
[2]Department of the Radio Engineering and Information Security, Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine

*Corresponding author (E-mail: s.nichyi@chnu.edu.ua)

ABSTRACT The article considers the possibility of using a key distribution method in a wireless sensor network by analogy with the quantum key distribution method. It is assumed that the wireless sensor network is created using hardware that allows the use of a predetermined N communication channels in the key distribution session, out of all possible ones (Nmin ≤ N ≤ Nmax). The initialization of the key distribution procedure is initiated by the master device/central node. At a certain time, the initiator switches to the transmission mode, and the slave devices to the reception mode. The master device generates k binary code packets (half-bytes, bytes), the number of which is predetermined. Each packet is transmitted over a randomly selected channel nTX from the range N. Slave devices receive the packet data over a randomly selected channel nRX from the range N. Based on the received packets, the master and slave devices generate a key for data exchange in a pair. The key generation algorithm is as follows: the slave device randomly selects a reception channel from consecutive k packets, of which there is a number of channel matches q, with reliable data (nTX = nRX). The remaining (k – q) received packets are unreliable, because they contain random data (channel mismatch nTX ≠ nRX). The slave device transmits to the master device a binary packet with a sequence of its own reception channel numbers. The encryption-decryption key for the master-slave pair is formed on the basis of q reliable packets, which are also determined by the master device from the received packet of the sequence of reception channels. The cryptographic stability of the generated key depends on the number of received packets q containing reliable data, a given number of channels N of the slave device of the network, as well as on the number of channels identified by a potential attacker. Methods for software and hardware implementation of this algorithm are considered, and an analysis of experimental results of using the proposed algorithm is carried out.

KEYWORDS wireless sensor networks, key distribution, random channel selection, random number generator, crypto protection.

I. INTRODUCTION

The use of wireless sensor networks (WSN) provides significant advantages during network deployment for the implementation of a specific functional application. At the same time, the operation of the embedded software of the WSN is subject to the condition of limiting computing and energy resources. In some specialized networks, there is a minimum service life of autonomous power supplies, which is two to three years (fire, security and other systems). Therefore, all WSN functionality is optimized during design for maximum energy efficiency of the network. Data encryption in the WSN is an important link in the policy of preserving data integrity, network operation, preserving its architecture. Key distribution in a network provides confidentiality and authentication of network components. Different key distribution methods have different energy efficiency, so this is taken into account when using a certain method in building a network. The article considers the possibility of using the key distribution method in a wireless sensor network by analogy with the quantum key distribution method using random selection of communication channels.

II. MODERN ISSUES OF KEY DISTRIBUTION IN WSN

Data transmission over the air by the WSN are vulnerable to various methods of influencing their functionality. Therefore, ensuring the security of WSN functionality is an important criterion for building these networks. The choice of key distribution algorithms is determined based on the conditions of limited energy resources of power sources and computing power. Therefore, the optimal method of key distribution in WSN is an important task when designing a network with a given functionality. The most numerous components of WSN are sensor devices (end devices (ED)) that transmit data to the central node (coordinator, access point (AP)). As a rule, the electrical power of the ED is provided by autonomous power sources. These sources must correspond to both the appropriate energy capacity and certain overall dimensions. As a rule, end-sensor devices do not need to have significant computing power. They perform a limited set of operations: measurement and transmission of measurement results. In sensor nodes, the main sources of energy consumption in the process of key distribution are [1]:

- computational costs – energy consumption that occurs when performing cryptographic operations, including key generation, hashing, encryption and signing;
- communication costs – energy required to transmit and receive messages between nodes. Data transmission requires more energy than local calculations;

- key storage – in large networks, each node can store dozens or hundreds of keys, which creates an additional load on the device's memory and energy consumption.

Today, there are many methods and algorithms for key distribution in computational sensor networks:

- Pre-key distribution [2]. The simplest and most energy-efficient approach, in which pre-generated keys are loaded into nodes before deployment. This method provides minimal energy consumption, since it does not require any calculations or additional communication during network operation. However, it is vulnerable to node compromise;

- Blom's scheme [3] – The method is based on linear algebra algorithms and allows nodes to calculate a shared key based on a pre-distributed matrix. Although the scheme provides better resistance to compromise, it requires additional calculations, which increases energy consumption compared to basic methods;

- SPINS, LEAP protocols [4]. These protocols implement symmetric encryption with local generation of session keys and are focused on energy efficiency. For example, the SPINS protocol (Security Protocols for Sensor Networks) uses only symmetric encryption and one-way hash functions, which allows to reduce energy consumption. LEAP (Localized Encryption and Authentication Protocol) uses four types of keys, which increases flexibility without significant energy load;

- Public-key cryptography methods and asymmetric cryptography, in particular RSA, have traditionally been considered unsuitable for WSN due to high energy costs. However, the use of elliptic curve cryptography (ECC) [5] allows to significantly reduce these costs. ECC provides a high level of security with a relatively small key length (for example, ECC-160 provides RSA-1024 level security), which reduces computational costs. Despite this, asymmetric schemes remain more energy-intensive than symmetric ones;

- blockchain-based key distribution methods [6]. Using blockchain technology WSN allows for a high level of trust and transparency. However, this model requires a significant amount of computation and communication, which makes it unsuitable for energy-constrained sensor nodes without special optimization.

The crypto-resistance and integrity of the network can be increased by combining several key generation algorithms. The use of specific key distribution algorithms in WSN depends on its purpose. In WSN, data exchange over radio channels can be used to obtain the initial key material, which, in combination with other methods described above, will allow the formation of encryption keys.

In quantum cryptography [7], the key distribution method proposed by Bennett and Brassard is well-known. One of the advantages of this method in key distribution is the randomness of the choice of the basis for transmitting and receiving a quantum of radiation. Alice (transmitter) (Fig. 1) generates a random sequence of bits. For each bit, she randomly chooses one of two types of bases for encoding a photon: rectilinear or diagonal. She sends photons to Bob (receiver) via a quantum channel (for example, an optical fiber). Bob measures the transmitted photons, randomly choosing the basis for receiving, and records the result. Alice and Bob, through an open classical channel, inform each other which bases were used for each bit, but not the bit values themselves. The bits obtained when the bases coincide are the basis for key formation.

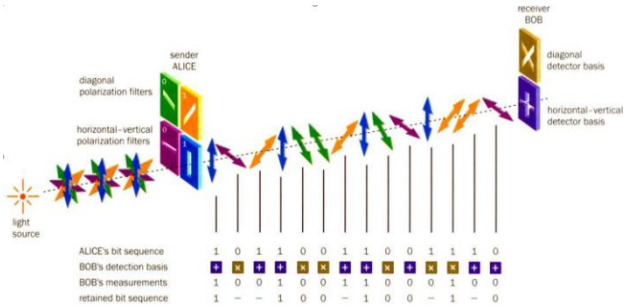

FIG. 1. Key generation algorithm in quantum key distribution.

If, instead of changing the bases, the data reception-transmission channels are changed in WSN to form the key, this will make it possible to partially use this algorithm in radio channel communication. In particular, in [8] the authors used the idea that in the process of agreeing on key questions, Alice and Bob randomly choose a channel for sending and listening, respectively. If they choose the same channel, the key information will be successfully received and Bob sends an acknowledgement (ACK) during the timeout period. Otherwise, there will be no acknowledgement. Alice chooses a new channel and repeats the process with new key material. The key is formed based on information that has an acknowledgement. The authors proposed this method for two wireless devices. Let the time t be required to transmit the key material, and the time τ to transmit the WSN signal. For k processes of key material matching in a WSN containing b ED, the key material formation time will be equal to

$$T = k(t + \tau b). \quad (1)$$

When periodically updating keys in the network, this time will significantly affect the energy efficiency of EDs, which must switch to the transmission mode kb once in each key generation process. When generating key material, it should be borne in mind that a smaller number of transmission-reception sessions leads to an increase in energy efficiency, and an increase in the amount of data for key generation increases the cryptographic strength of the network. The purpose of our work is to study and evaluate the possibility of creating a key distribution method in a WSN by analogy with the quantum key distribution method while reducing the time for generating key material.

III. DESCRIPTION OF THE KEY DISTRIBUTION ALGORITHM IN A WIRELESS SENSOR NETWORK

Radio channel communication uses the physical properties of the channel. For example, in a WSN based on the IEEE 802.15.4 for 2.4 GHz standard, blocks of samples of 16 channels [9] can be used. Different channels mean

not only the frequency range, but also other data transmission parameters (modulation parameters, spectrum spreading, scrambling, etc.).

We propose an algorithm [10] for key distribution in a WSN of a star or point-to-point structure. The WSN is created using hardware that allows the use of a predefined N communication channels, key distribution sessions, from all possible ones ($Nmin \le N \le Nmax$). The initialization of the key distribution procedure is initiated by the master device AP. At a certain time, the initiator switches to the transmission mode, and the slave devices ED to the reception mode. The master device generates k binary code packets (half-bytes, bytes), the number of which is predetermined. Each packet is transmitted on a randomly selected channel n_{TX} from the range N. The slave devices receive the data packets on a randomly selected channel n_{RX} from the range N. Based on the received packets, the master and slave devices generate a key for data exchange in a pair. The key generation algorithm is as follows: the slave device randomly selects a receive channel from consecutive k packets, of which there is a number of channel matches q, with reliable data ($n_{TX} = n_{RX}$). The remaining (k–q) received packets are unreliable, because they contain random data (channel mismatch $n_{TX} \neq n_{RX}$). The slave device transmits a binary packet with a sequence of its own reception channel numbers to the master device without information about the received data. This packet is transmitted over a predetermined radio channel and time interval for each ED of the network. The encryption-decryption key for the master-slave pair is formed based on the primary key materials from q authentic packets, which are also determined by the master device from the received packet of the receive channel sequence. Of course, unlike quantum key distribution, in this case it is impossible to determine the fact of "eavesdropping". An example of key formation, according to the proposed method, is presented in Table 1.

The theoretical probability of coincidence of randomly selected transmit-receive channels can be calculated using the binomial distribution formula [9].

$$P(q) = \frac{k!}{q!(k-q)!} \cdot \left(\frac{1}{N}\right)^q \cdot ((N-1)/N)^{(k-q)}. \qquad (2)$$

The results of calculations of the probability of the number of channel coincidences q for different values of k, N for the condition $1 \le q \le 3$ are shown in Fig. 2.

The highest probability of three coincidences of reception-transmission, with the number of channels N more than 3, is expected after five transmissions. The real randomness of the coincidence of channels during key formation is determined by the algorithms for generating channel numbers, which are based on the use of random variables implemented by software and hardware means of processor systems. For example, using the *rand()* function of the compiler of the C/C++ programming language of microcontrollers. The channel number refers not only to the transmission and reception frequency, but also to changes in other physical channel parameters, for example in the LoRa standard [11].

TABLE 1. Example of forming primary key material between AP and ED when N=4, k=8, q=2.

Transfer stage from AP to ED			
Transmission channels AP	**Data in packages from the AP**	**Reception channels ED**	**Data for creating an encryption key**
n0	Data1	n3	
n3	Data2	n0	
n1	Data3	n1	Data3
n2	Data4	n3	
n2	Data5	n2	Data5
n0	Data6	n1	
n1	Data7	n0	
n0	Data8	n1	

Transfer stage from ED to AP		
Transmission channels ED	**Reception channels AP**	**Data in packages from ED to AP**
n_N	n_N	n3 n0 n1 n3 n2 n1 n0 n1

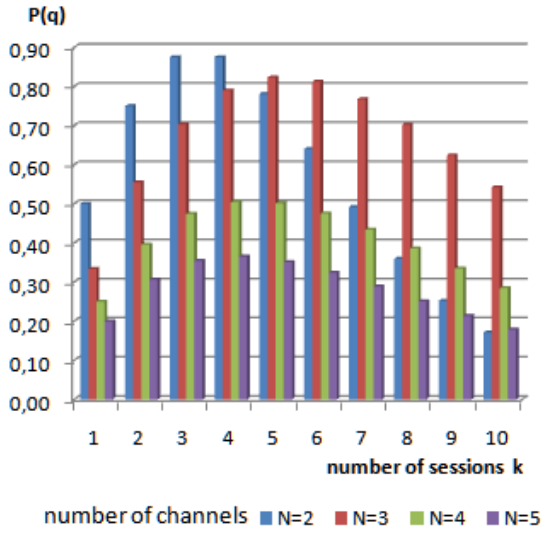

FIG. 2. Graph of the probability of channel coincidence q depending on the values of k and N for the condition $1 \le q \le 3$.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

This key distribution algorithm was tested using the WSN built on GB2530 [12] modules, which use the CC2530 SoC [13]. The embedded software was developed using the MPLAB X IDE v5.05 package, and Texas Instruments Packet Sniffer was used to analyze the transmission packets. To test the method, four communication channels were used according to Table 1 and Fig. 1.

To generate four random channel numbers, using the C/C++ compiler, we used the operator:

chenal[i] = rand()%4;

Based on these numbers, four channel numbers are formed. Partial results obtained using the above operator are presented in columns 2 and 3 of Table 2 (column 2 shows the results of channel generation by the AP) (yellow cells show the coincidence of channel numbers between the AP

and the ED). According to the algorithm of the *rand()* function, the same sequences of receive-transmit channels will be obtained in all devices. The C/C++ compiler sets the initial number relative to which the "random" value is formed. The last "random" value obtained is the basis for obtaining the next "random" value. This synchronicity will be observed only when the network components are synchronized (same power-on time, same sequential reception-transmission sessions). If the equipment loses synchronicity, the numbers of the generated channel numbers are also shifted. If five packets are used for a key generation session, when shifted, as a result of skipping one session, in most cases the key can be generated from only one data packet. This can be seen if the third column is shifted relative to the second by five rows down. To form a reliable key, a larger number of data packets received from synchronized channels is required. It should also be borne in mind that increasing the number of packets in a key generation session increases energy consumption.

The value of the series of generated random numbers can be changed by using the *srand()* function. This function sets the initial value of the number that is the basis for obtaining a random number using the *rand()* function. This will allow for each key generation session to obtain higher probabilities of matching the receive-transmit channels. Such a value can be obtained from approximately the same parameters of the wireless network, for example, the system time, the last value of the checksums (CRC) of the transmission sessions of the end devices. If you use a set of operators:

```
srand(data);
chenal[i] = rand()%4;
```

then the same values of data for the *srand()* function of all network devices provide a high probability of matching the channels during the key generation session. For example, Table 2 shows the channel numbers (from the set 0,1,2,3) obtained using *rand()* and *rand()-srand()*. The data given in Table 2 are purely illustrative and are presented as an example of a practical implementation of the algorithm, without claiming their universality. The results are predictable, based on the algorithm for obtaining random numbers of programming language compilers for creating embedded WSN software.

To generate a random packet transmission and reception channel for key formation, it is also possible to use the hardware of the WSN components. For example, to generate a random number, we used the *RFRND* register of the SoC CC2530 radio frequency core [13].

The value of the *IRND* bit of the *RFRND* register is formed based on the noise of the radio frequency range. Considering that the WSN components are alternately in the reception or transmission mode, forming such a list of random numbers does not require additional energy and machine time. Randomness tests show good results [13], but there is a constant component of converting the radio signal into binary code. If we interpret the obtained values as integers from 0 to 255, the average value is 127.6518, out of 20 million values.

TABLE 2. Sequence of transmit and receive channel numbers obtained when using the function rand().

1	AP 2	ED1 3	ED2 4	ED3 5	ED4 6	ED5 7
			srand (25)	srand (567)	srand (2467)	srand (3245)
1	2	2	2	3	1	0
2	2	2	2	0	2	3
3	1	1	1	1	1	1
4	3	3	3	0	3	1
5	3	3	3	1	1	3
6	3	3	3	1	2	2
7	2	2	2	0	0	2
8	3	3	3	3	3	3
9	0	0	0	2	1	3
10	2	2	2	3	1	0
11	1	1	1	0	1	0
12	3	3	3	0	2	2
13	0	0	0	2	2	3
14	0	0	0	2	3	3
15	1	1	1	1	2	2
16	3	3	3	3	2	2
17	1	1	1	1	3	3
18	3	3	3	1	0	2
19	1	1	1	2	3	3
20	2	2	2	1	3	0

TABLE 3. Sequence of transmit and receive channel numbers obtained based on radio band noise.

	AP	ED1	ED2	ED3	ED4
t+1	01	02	03	01	02
t+2	03	00	02	03	01
t+3	03	00	01	03	03
t+4	02	01	02	03	02
t+5	01	02	01	03	00
t+6	02	01	03	02	00
t+7	00	03	03	01	00
t+8	01	03	02	02	01
t+9	02	02	00	01	03
t+10	00	01	00	02	02
t+11	00	03	00	00	03
t+12	00	03	00	00	02
t+13	00	03	01	01	01
t+14	00	02	03	02	03
t+15	00	00	02	01	03
t+16	00	01	00	03	02
t+17	00	03	01	02	00
t+18	00	03	03	01	01
t+19	00	02	02	03	02
t+20	01	01	01	03	00

An example of a C code fragment for obtaining a random channel from four possible ones (from 0 to 3), which we used, looks like this:

```
rendbyte=(rendbyte << 1)|( 0x01 & RFRND);
rendchanell=rendbyte%4;
```

The first operator forms the value of the number in the variable rendbyte based on the read *IRND* bit of the *RFRND* register. A logical left shift operation and logical addition of the *IRND* bit value are used. The second operator calculates the channel number based on the remainder operation.

The results of the received sequences of transmission reception channels from some time point t, components of the WSN are presented in Table 3. Column AP shows the transmission channels of the base device, the remaining columns ED show the reception channels (the data do not claim to be general or exhaustive). Yellow cells show the coincidence of channel numbers between AP and ED. The columns AP and ED in Table 3 can be interchanged, since the roles of the devices are chosen randomly. For example, ED2 is assigned the role of AP and AP is assigned the role of ED2. Unlike the case of using the rand() function, hardware generation of a random channel number is less predictable.

V. CONCLUSION

The results show that the proposed method of distributing primary key material can be used in WSN. Unlike the key distribution method in [8], the proposed method makes it possible, as follows from Eg.1, to conduct sessions of obtaining material for creating a key k times faster. The probability of listening to the data exchange channel by an outside attacker is determined by $p=N/(Nmax-Nmin)$. If we ignore the presence of the attacker with receiving equipment to listen to all possible transmission and reception channels that the network coordinator can use to form a key, then the use of different software encryption algorithms based on the data obtained in the sense of distributing key material will provide greater network resistance to hacking. When using software algorithms for generating random numbers on a data transmission channel, to increase the probability of channel coincidences, it is advisable to use the srand() function, the argument of which will be the same for all network devices (for example, system time). This will allow to increase the number of channel coincidences with smaller numbers of reception-transmission sessions and, accordingly, data on the basis of which will be formed and save energy resources of sensors (ED).

Using hardware-based data channel acquisition algorithms, unpredictable sampling and low coincidence of the receive-transmit channels are observed. In this case, using hash functions to generate a key based on the order of channel number sampling can allow obtaining an encryption key without the receive-transmit data..

Various algorithms for selecting the start time of a data exchange session, the list of data exchange channels, the number of packets in a session, methods for generating "random" channel numbers, etc., with the simultaneous use of software tools for generating keys described in section 2, contribute to increasing the cryptographic strength of the WSN.

AUTHOR CONTRIBUTIONS

S.N. – idea, research, writing (reviewing and editing); P.S. – research methodology, writing (reviewing and editing); B.N. – hardware and software development, research, resources, writing – preparation of the original draft.

COMPETING INTERESTS

The authors declare no competing interests.

REFERENCES

[1] H. Zhang, M. Zhang, T. Qin, W. Wei, Y. Fan, and J. Yang, “An Energy Consumption Optimization Strategy for Wireless Sensor Networks via Multi-Objective Algorithm,” *Journal of King Saud University – Computer and Information Sciences*, vol. 36, no. 1, p. 101919, 2024.

[2] L. Eschenauer and V. D. Gligor, “A Key-Management Scheme for Distributed Sensor Networks,” in *Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS 2002)*, ACM, Nov. 2002, pp. 41–47.

[3] S. N. F. M. A. Antony and M. F. A. Bahari, “Implementation of Elliptic Curves in the Polynomial Blom Key Pre-Distribution Scheme for Wireless Sensor Networks and Distributed Ledger Technology,” *Journal of Sensor and Actuator Networks*, vol. 12, no. 1, Art. no. 15, 2023.

[4] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, “SPINS: Security Protocols for Sensor Networks,” *Wireless Networks*, vol. 8, no. 5, pp. 521–534, 2002.

[5] U. Gulen, A. Alkhodary, and S. Baktir, “Implementing RSA for Wireless Sensor Nodes,” *Sensors*, vol. 19, no. 13, p. 2864, 27 Jun. 2019.

[6] S. Md. Habibullah, S. Alam, S. Ghosh, A. Dey, and A. De, “Blockchain-Based Energy Consumption Approaches in IoT,” *Scientific Reports*, vol. 14, no. 1, Art. no. 28088, 15 Nov. 2024.

[7] C. H. Bennett and G. Brassard, “Quantum Cryptography: Public Key Distribution and Coin Tossing,” in *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing*, Bangalore, India, 1984, pp. 175–179.

[8] B. Zan and M. Gruteser, “Random Channel Hopping Schemes for Key Agreement in Wireless Networks,” in *Proc. IEEE 20th Int. Symp. Personal, Indoor and Mobile Radio Communications (PIMRC)*, Tokyo, Japan, Sept. 2009, pp. 1–5.

[9] IEEE, *IEEE Standard for Low-Rate Wireless Networks (IEEE 802.15.4-2020)*, 2020.

[10] P. M. Shpatar, B. S. Nichyi, and S. V. Nichyi, “Key Distribution in a Wireless Sensor Network Using the Random Channel Selection Method,” in *X International Scientific-Practical Conference Physical and Technological Problems of Transmission, Processing and Storage of Information in Infocommunication Systems*, Chernivtsi, Ukraine, 15–17 May 2025, p. 163.

[11] P. Shpatar, B. Nichyi, and S. Nichyi, “Study of energy efficiency of LoRa technologies in wireless monitoring networks,” *International Scientific-technical journal «Measuring and computing devices in technological processes»*, 2024, Issue 3, pp. 166–174.

[12] “CC2530 Zigbee Low Power Module GB2530-L,” *Bossgoo.com*, Xiamen Gban Electronic Technology Co., Ltd. [Online]. Available: https://www.bossgoo.com/product-detail/cc2530-zigbee-low-power-module-gb2530-44037167.html. Accessed: 1 Feb. 2026.

[13] Texas Instruments, *CC2530F32, CC2530F64, CC2530F128, CC2530F256: A True System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications*, SWRS081B, Apr. 2009, revised Feb. 2011.

Serhii Nichyi

Ph.D., Associate Professor of the Department of Electronics and Power Engineering, Chernivtsi National University. Scientific interests: technology of thin semiconductor films, automation of measurements on primary converters and use of wireless sensor networks for automation systems.

ORCID ID: 0000-0003-2662-9694

Bohdan Nichyi

He graduated from Chernivtsi National University with a master's degree in "Electronic Communications and Radio Engineering". Currently a postgraduate student of the Department of Radio Engineering and Information Security. Research interests include the security of telecommunication networks and systems.

ORCID ID: 0009-0001-4855-2053

Petro Shpatar

Ph.D. (Engineering Sciences) Associate Professor of Department of Radio Engineering and Information Security. Director of the Educational and Scientific Institute of Physical, Technical and Computer Sciences. Research interests: pseudorandom sequence generators; encryption information Author of nearly 60 publications.

ORCID ID: 0000-0003-4088-1458

Розподіл ключів у бездротових сенсорних мережах методом випадкового вибору каналу

Сергій Нічий[1,*], Петро Шпатар[2], Богдан Нічий[2]

[1]кафедра електроніки і енергетики, Чернівецький національний університет імені Юрія Федьковича, Україна
[2]кафедра радіотехніки та інформаційної безпеки, Чернівецький національний університет імені Юрія Федьковича, Україна

*Автор-кореспондент (Електронна адреса: s.nichyi@chnu.edu.ua)

АНОТАЦІЯ В статті розглядається можливість застосування методики розподілу ключів в безпровідній сенсорній мережі по аналогії із методом квантового розподілу ключів. Передбачається, що безпровідна сенсорна мережа створена з використанням апаратного забезпечення яке дозволяє використовувати, наперед визначених N каналів зв'язку в сеансі розподілу ключів, з усіх можливих (Nmin ≤ N ≤ Nmax). Ініціалізацію процедури розподілу ключів розпочинає ведучий пристрій/центральний вузол. У визначений час ініціатор переходить в режим передачі, а ведені пристрої в режим прийому. Ведучій пристрій формує k посилок двійкового коду (пів-байт, байт) кількість яких визначена наперед. Кожна посилка передається по випадково вибраному каналу nTX з діапазону N. Ведені пристрої приймають дані посилки по випадковому вибраному каналі nRX з діапазону N. На основі отриманих посилок ведучі та ведені пристрої формують ключ для обміну даними в парі. Алгоритм формування ключа є наступним: ведений пристрій вибираючи випадково канал прийому із послідовних k посилок з яких є кількість співпадінь каналів q, з достовірними даними (nTX = nRX). Решта (k – q) прийнятих посилок є недостовірні, тому що містять випадкові дані (неспівпадіння каналів nTX ≠ nRX). Ведений пристрій передає ведучому пристрою двійкову посилку із послідовністю номерів каналів власного прийому. Ключ шифрування-дешифрування для пари ведучій-ведений формується на основі q достовірних посилок які також визначає ведучий пристрій із отриманої посилки послідовності каналів прийому. Криптографічна стійкість згенерованого ключа залежить від кількості прийнятих посилок q, що містять достовірні дані, заданої кількості каналів N веденого пристрою мережі, а також від кількості каналів ідентифікованих потенційним зловмисником. Розглянуто методи програмної та апаратної реалізації даного алгоритму, та проведено аналіз експериментальних результатів використання запропонованого алгоритму.

КЛЮЧОВІ СЛОВА бездротові сенсорні мережі, розподіл ключів, випадковий вибір каналу, генератор випадкових чисел, криптозахист.