Approaches to the application of homomorphic encryption in sensor networks

L K Babenko, E A Tolomanenko
Southern Federal University, Taganrog, Chekhov str., 2, 347922, Russia
E-mail kat.tea@mail.ru

Abstract. This paper describes the basic principles of the sensor network. The article characterizes the most common types of attacks on wireless sensor networks. The most commonly used protocols of sensor networks are presented. Special attention is paid to schemes of partially homomorphic encryption and fully homomorphic encryption schemes. An approach to the implementation of sensor network protection using the joint use of a fully homomorphic encryption algorithm and a symmetric block encryption algorithm is proposed.

Keywords: sensor network, attacks on a sensor network, cryptography, homomorphic encryption

1. Introduction
The development of IT technologies is gaining momentum over time, cyber-physical systems are used almost everywhere: security systems, smart home systems, diagnostics of various equipment in production, collection of environmental indicators, healthcare, the military-industrial complex, etc. Wireless sensor networks are often the obvious basis for such systems. Sensor networks are a wireless network distributed in space, which consists of many small sensor devices - sensors. Sensors have the capabilities and functionality to receive and transmit information. All distributed sensors are used to record and track readings from their environment, or to collect all received data in one place. For example, touch sensors can collect information about environmental conditions, including temperature, humidity, noise level, pollution, wind speed, etc.

However, wireless sensor networks have a number of advantages and disadvantages. When it comes to value, the sensors are fault tolerant, scalable, mobile, able to withstand harsh environmental conditions, and are easy to use. Within the range of this network, you can access the device at any time. The disadvantages of this network are the low power consumption of the devices, low performance due to the small size of each sensor, and also the small amount of memory. Common methods of protecting computer networks are not suitable for protecting wireless sensor networks, so all this leads to the development of special approaches for protecting sensor networks that will be relevant specifically for them, for example, using homomorphic encryption schemes.

An example of building a sensor network is shown in Fig. 1. The nodes of this network are low-cost from an energy point of view, but at the same time this is the multifunctional sensors. Also, sensor networks often include base stations that receive information from a group of sensors associated with it.
The most famous protocols on the basis of which sensor networks are capable of functioning:
- MQTT is an open data exchange protocol in conditions of a small amount of data transferred and limited bandwidth;
- AMQP - used to exchange data at the application level, between system components through a broker that routes packets;
- MAVLink is a protocol that describes the interaction between ground systems and components of the network - various small unmanned aerial vehicles, ground vehicles, water vehicles, and drones.

2. Attacks on sensor networks
Sensor networks have a number of problems related to the security of the information they transmit. These problems arise due to the disadvantages listed above, which are the weak points of these networks. Since any attacker knows in advance that the nodes of the sensor network collect and transmit any information to each other, he can make attempts to invade such a network. The most famous attacks on sensor networks are the following:

1. Denial of Service – Bringing a system to failure by transmitting a huge amount of data requests. Thus, the entire attacked transmission channel will be loaded, the nodes will become unavailable, and the resources of the sensors exposed to the attack will be depleted [1].
2. Creation of a fake route – Attackers register a new sensor in the sensor network implementing this attack, which is beneficial for surrounding sensors using fake routing data [2].
3. Creation of wormholes - In this attack, attackers work together to create a new fake node and transmit this data to the destination over a high-speed connection. Due to the high speed of the transmission channel, all nodes will start transmitting information through fake nodes, and the attacker will eventually receive all the data [3].
4. Passive collection of information - an attacker using additional technical means, in the form of a specialized antenna, collects information transmitted in the network, which he can then use to attack the network.

3. Types of homomorphic encryption
Homomorphic encryption is an encryption method that allows you to perform mathematical operations, namely addition and multiplication, with ciphertexts and obtain an encrypted result that will correspond to the result of mathematical operations on the plaintext. Homomorphic encryption allows you to perform various operations in an untrusted environment without revealing confidential data for each of the operations [4].

There are two types of cryptosystems: fully homomorphic and partially homomorphic. Fully homomorphic schemes support both addition and multiplication mathematical operations, that is, the properties of homomorphism are performed on these operations. A partially homomorphic cryptosystem allows performing only one of the above operations — either addition or multiplication. Partially homomorphic cryptosystems are represented by the following schemes.
The product of two $m + M_p = g$, $E$, $m + u = r$, and $E = a + y$, $r = M_x$.

The computational complexity of the homomorphic encryption scheme is due to the fact that in order to encrypt a $K$-bit message, it is necessary to carry out operations in the amount $O(K(\log_2 N)^2)$. The original text will increase in $\log_2 N$ times when encrypted, because each bit of the original text corresponds to a $\log_2 N$ bit of the ciphertext. The semantic stability of this scheme, due to the resistance to an attack based on a selected source text, lies in the fact that an adversary with computational polynomially limited resources, as well as knowing the ciphertext and its length corresponding to a certain plaintext, will not extract any useful information.

Cryptosystem Benalo [6] is a modified Goldwasser and Micali scheme described above. The main difference is that this scheme allows one-time encryption of an entire block of data, in contrast to the Goldwasser and Micali scheme, where each bit of the plaintext is encrypted separately. In the Benalo cryptosystem, the homomorphism properties are also performed over the addition operation:

$$E(M_1) \cdot E(M_2) = (x^{M_1}u_1^r)(x^{M_2}u_2^r) = x^{M_1+M_2}(u_1u_2)^r = E(M_1 + M_2 \mod r)$$

where $r$ is the block size. The computational complexity of this kriptoskhemy: $O(R(\log n)^2)$. Subject to the use of preliminary calculations, the time complexity of decryption $O(\sqrt{r})$. The robustness of Benalo's scheme is due to the computationally difficult task of high-degree residues: it is computationally difficult to find a plaintext, even with information about the block size, the magnitude of the modulus, with an unknown factorization and ciphertext.

PAYE's cryptosystem [7] invented by P. Paillier, is a public-key cryptosystem, probabilistic, strength due to the complexity of factorization of a composite number, which is the product of two primes. The homomorphism of this scheme is also performed over the addition operation. In the PAYE cryptosystem, the multiplication of two ciphers will be decoded as the sum of their respective plaintext:

$$D(E(m_1, r_1) \cdot E(m_2, r_2) \mod n^2) = m_1 + m_2 \mod n.$$ 

Multiplying the ciphertext by $g^{m_2}$, we get the sum of the corresponding plaintexts:

$$D(E(m_1, r_1) \cdot g^{m_2} \mod n^2) = m_1 + m_2 \mod n,$$

where $m_1$ and $m_2$ are plaintext, $(n, g)$ is a pair of numbers forming the public key, $r_1$ and $r_2$ are random numbers, $E$ is the encryption function, and $D$ is the decryption function.

Cryptosystem of Damgard and Yurik [8]. This system is interpreted as a generalization of the above-described PAYE cryptosystem for large modules for use in a wider range of tasks. The homomorphism property holds over the addition operation, the system is homomorphic with respect to addition to $Z_n^*$:

$$E(M_1) \cdot E(M_2) = E(M_1 + M_2 \mod n^2)$$

where $n$ is a - module and $s$ - natural number. As for the strength of the Damgard and Yurik cryptosystem, it is based on the assumption that it is difficult to calculate the root of the remainder modulo, as it is implemented in the above-described PAYE system.

Cryptosystem Okamoto – Uchiyama [9]. This system is probabilistic, based on a logarithmic function, which is defined over the multiplicative group and using large primes. Additive homomorphism of the system due to the fact that $M_1 + M_2 < p$ performed as follows:

$$E(M_1) \cdot E(M_2) = (x^{M_1}r_1^x)(x^{M_2}r_2^x) \mod n = x^{M_1+M_2}(r_1r_2)^x \mod n = E(M_1 + M_2)$$

The stability of the Okamoto – Uchiyama system is based on the complexity of the factorization problem for the number $n$ and requires $O(\log_3 n)$ bitwise operations.

Fully homomorphic systems are a much more rare form of homomorphic encryption. As previously stated, Fully Homomorphic Encryption can support simultaneous addition and multiplication operations, allowing more computations to be performed on encrypted data. In 2009, an IBM employee,
Craig Gentry, proposed the first cryptographically strong asymmetric fully homomorphic encryption scheme based on ideal lattices [10]. In recent years, many works have appeared in which various modifications and optimizations of this scheme have been proposed. The homomorphic property of the simplified Gentry scheme is to obtain the sum of texts $m_1 + m_2$, you only need to add the ciphertexts and perform the decryption operation:

$$D(E(m_1, pk) + E(m_2, pk)) = D \left( m_1 + m_2 + \sum_{i=0}^{n} C_ipki \right) = D(m_1 + m_2 + C'_1sk + C'_2n)$$

where $pk$ and $sk$ are the public and private keys, respectively.

In 2012, Gentry showed how you can perform self-correction of ciphers [11], and then showed how you can use a fully homomorphic encryption scheme in conjunction with the AES encryption algorithm [12]. The idea proposed by Gentry is as follows. For data encrypted first with a symmetric cipher and then with a homomorphic cipher on top, it is possible to decrypt with a symmetric system using the encryption key without removing the homomorphic encryption. To do this, it is necessary to represent all operations of the decryption algorithm in the form of simple Boolean functions, namely, those functions that allow the selected type of completely homomorphic cipher to be performed.

4. Development of a homomorphic encryption scheme using a symmetric algorithm

Based on Gentry's ideas and the functioning of sensor networks, as well as existing implementations and modifications of fully homomorphic encryption schemes, an approach to information protection in sensor networks was developed with the joint use of a fully homomorphic scheme and a symmetric encryption algorithm.

The Kuznechik encryption algorithm - GOST R 34.12-2015 [13] was chosen as a symmetric encryption algorithm. As you know, Kuznechik uses S-block substitution in his transformations, which is the most difficult. The replacement S-block, in turn, can be represented in the form of Zhegalkin polynomials, which will reduce the complexity of this task.

The development of a data transfer algorithm based on the selected encryption methods implies interaction between the sensor nodes of the network and the base station, where the most resource-intensive operations will be performed at the base station, which is not limited in the battery supply, and low-cost calculations and the exchange of processed data itself will be performed directly by the sensor nodes of the network. Thus, all complex encryption operations reproduced by the Kuznechik algorithm and the homomorphic Gentry scheme will be implemented at the base station. And the nodes will then be able to transmit homomorphically encrypted data without much impact on the energy and processor components.

In view of the fact that symmetric encryption algorithms are energy-intensive during the execution of encryption and decryption operations, it becomes necessary to develop a method for implementing the symmetric algorithm in the fastest possible way in order to reduce the computational and associated time costs of the sensor nodes and the base station.

5. Conclusion

This paper contains a description and application of sensor networks, which have become the most relevant main components in the construction of cyber-physical systems. The most common types of attacks on this wireless sensor network are considered. The most well-known partial homomorphic encryption schemes used in modern cryptographic systems have been described. The analysis of existing fully homomorphic cryptosystems based on the idea of Gentry is carried out. An approach was developed for the joint use of a homomorphic encryption scheme and a symmetric cryptoalgorithm in order to protect data transmitted in sensor networks. Further work will be devoted to the working and development of an approach to the joint use of symmetric and homomorphic encryption algorithms.

6. Acknowledgment
The reported study was funded by RFBR, project number 19-37-90148.

References
[1] What is a DoS attack? (Binance Academy). URL: https://www.binance.vision/ru/security/what-is-a-dos-attack
[2] Sharmila S and Umamaheswari G 2011 Detection of Sinkhole Attack in Wireless Sensor Networks Using Message Digest Algorithms International Conference on Process Automation, Control and Computing (Coimbatore) pp 1-6
[3] Pandey and Rajshree 2014. Wormhole Attack in Wireless Sensor Network International Journal of Computer Networks and Communications Security vol 2 pp 22-26
[4] Sen J 2013 Homomorphic Encryption: Theory and Application Theory and Practice of Cryptography and Network Security Protocols and Technologies pp 31
[5] Goldwasser and Micali S 1982 Probabilistic encryption and how to play mental poker keeping secret all partial information Proceedings of the 14th ACM Symposium on the Theory of Computing. pp 365–377
[6] Benaloh J Verifiable secret-ballot elections URL: http://www.cs.yale.edu/publications/techreports/tr561.pdf
[7] Paillier P 1999 Public-key cryptosystems based on composite degree residuosity classes Advances in Cryptology - EUROCRYPT’99 - Vol. 1592 of Lecture Notes in Computer Science. pp 223–238
[8] Damgard I, Jurik M A 2001 generalization, a simplification and some applications of Pailliers probabilistic public-key system Vol. 1992 of Lecture Notes in Computer Science pp 119-136
[9] Okamoto T and Uchiyama S 1998 A new public-key cryptosystem as secure as factoring Advances in Cryptology - EUROCRYPT ’98 - Vol. 1403 of Lecture Notes in Computer Science pp 308-318
[10] Gentry C Fully 2009 Homomorphic encryption using ideal lattices Proceedings of the 41st ACM symposium on theory of computing (STOC) (Bethesda, Maryland, just outside Washington, DC, USA) pp 169-178
[11] Brakerski Zvika Gentry Craig and Vaikuntanathan Vinod 2012 Fully homomorphic encryption without bootstrapping In Innovations in Theoretical Computer Science (ITCS’12), URL: http://eprint.iacr.org/2011/277
[12] Gentry C and Halevi S 2012 Smart NP Homomorphic Evaluation of the AES Circuit CRYPTO 2012 pp 850-867
[13] GOST R 34.12–2015 Information technology. Cryptographic information protection. Block ciphers URL: https://tc26.ru/standarts/natsionalnye-standarty/gost-r-34-12-2015-infomatsionnaya-tehnologiya-kriptograficheskaya-zashchita-informatsii-blochnye-shifry.html