Location-Based Protocol for the Pairwise Authentication in the Networks without Infrastructure

Sergey Nesteruk, Sergey Bezzateev
Saint-Petersburg State University of Aerospace Instrumentation
Saint-Petersburg, Russia
nesterus@protonmail.com, bsv@aanet.ru

***Abstract*—In this paper, we consider security issues arising in the development of the wireless networks without infrastructure, with the rapidly changing composition of the elements of such a network The LEAP Initial Protection (LEAP-IP) protocol proposed, which closes the vulnerability of the LEAP at the network initialization stage. Advanced LEAP-IP protocol allows to resist attacks on the radio channel, physical attacks on the device, and is energy efficient, that is especially important for devices with a limited power resource. Also, a classification of self-organizing networks and some variants of using the proposed pairwise authentication protocol is presented.**

I. Introduction

Since the advent of the Internet, telecommunication technologies have changed and greatly improved [1]. Now we are witnessing a new round of development, in which more and more content is not created by people, but is generated by devices, which is a logical continuation of automation of all spheres of human activity [2]. This paradigm is called the Internet of things (IoT) and was first announced by the inventor Kevin Ashton in 1999 [3].

The rapid growth in the popularity of the IoT today is caused by a sharp drop in prices for sensors and microcontrollers [4].

IoT systems are used everywhere [5]. It is expected that in 2018, the world spending on the IoT will reach $770 bln. Industrial production (Industrial Internet), transport (Machine Internet) and utilities (Environmental Monitoring) are foremost interested in such technologies [6].

This has given rise to many small companies offering their solutions designed to help staying safe in a rapidly changing environment. However, many large giants also devote their energies to researching in this field [7].

Security issues are the principal obstacle to the development of the IoT. If we do not focus on the security now, users will lose confidence in such solutions [8]. At the same time, security in the field of the IoT has its own specifics:

- IoT devices are very diverse, and therefore they may have different potential vulnerabilities, which makes traditional endpoint security models impractical.
- Devices have a limited battery resource.
- Most devices must work in real time.
- Devices are usually developed for a long life cycle. In this case, updating the software of devices is often a difficult task.

Devices are usually deployed in unattended environments, which increases the risk of node capture and complicates the diagnostics [9].

An important issue in the development of IoT systems is to ensure the safe transfer of data at the device level. We can consider this level as a wireless sensor network. A wireless sensor network (WSN) is a self-organizing system consisting of low-power nodes connected with a radio channel, which can act either as the passive sensors for data collection or as the actuators. In a system constructed this way, the devices must communicate with each other and react to changes in the environment so that the assigned task is carried out [10].

The purpose of our work is to find the efficient algorithm for devices authentication in the networks with the lack of infrastructure (ad hoc like networks).

In this paper, we will look at the prospective topologies of sensor networks in Section II and the existing authentication protocols in Section III. In Section IV, we will look at the identified vulnerability of the LEAP protocol that appears during the device initialization phase, and we will offer two alternative ways to close this vulnerability in Section V. In Section VI, we will present steps of the authentication protocol for the devices after their initialization. We'll describe the widespread attacks on sensor networks and show how the proposed protocol allows them to be avoided or mitigated in Section VII, and we will describe the features of using the proposed protocol in Section VIII.

II. Sensor network topology

To date, the most common topology, which is used by developers of the IoT systems, as well as platform developers, such as [11-15], is the «star». It implies the presence of one base station, with which all subscribers are connected. This approach is the easiest to implement, but often it is not the most effective, so the various types of self-organizing networks are gaining in popularity.

A self-organizing network is a network with a variable decentralized infrastructure [16]. Among the advantages of such networks, we would note:

- self-organization;

- self-recovering;
- low required power consumption;
- simple expandability;
- high coating density.

Among disadvantages, we would note:

- network complexity;
- overhead for the network maintaining;
- relay delay;
- a high power consumption of repeaters.

With the development of technologies of self-organizing networks, there has also been confusion in their classification. Users began to use the term "ad hoc" to denote the direct connection of two computers, one of which was an access point and provided access to the Internet [17]. In most cases, people say "mesh", "ad hoc", "mobile ad hoc", etc., implying that this is the same thing. Nevertheless, there are fundamental differences between them. Let us consider some types of self-organizing networks and their basic properties in order of increasing the provided capabilities and, correspondingly, increasing the complexity of their design.

A. Mesh networks

Mesh networks are mesh radio networks consisting of fixed routers that create a wireless backbone and a coverage area for mobile or fixed users with access to one of the routers. Mesh networks are based on the "star of stars" topology and have random connections of support nodes [18]. In Fig. 1 lines indicate the wireless connections of devices.

Because of the hierarchical structure, such networks are easy to design. That is why today they are most common among self-organizing networks and are successfully used in communication systems and sensor networks. However, for the same reason they are less reliable because if one network node fails, all lower-level devices bound to it will also be unavailable [17].

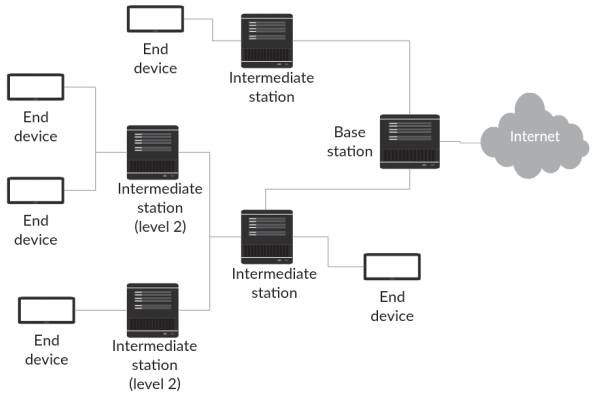

Fig. 1. An example of mesh network nodes connection

The main properties of mesh networks are:

- wireless;
- dynamic.

For wireless communication we usually use standards such as IEEE 802.11 Wi-Fi for local and city networks, IEEE 802.15.1 Bluetooth for home systems, IEEE 802.15.4 Zigbee for sensors.

Under the dynamic nature of the network we mean that it is configured itself, without human intervention. In this case, it can require also control or statistical information between the nodes participating in the organization of the network for receiving and transmitting data (for example, for balancing the load and sending information about any network topology changing) [19].

B. Ad hoc networks

The expression "ad hoc" came from Latin and translates as "for this case". Ad hoc networks are radio networks with random stationary subscribers, realizing completely decentralized control in the absence of base stations or support nodes. The topology of such networks is fixed and has an accidental connection of nodes (Fig. 2) [18].

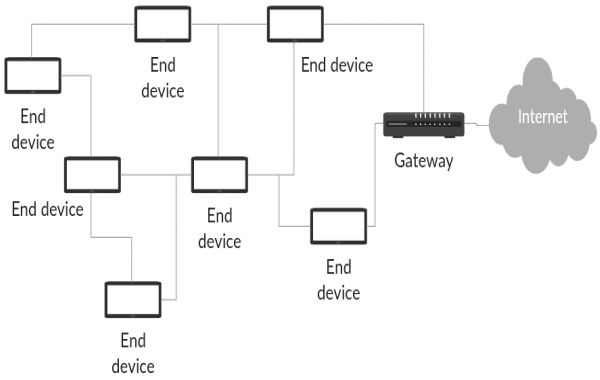

Fig. 2. An example of ad hoc network nodes connection

An important feature of such networks is that the nodes of such a network are independent of each other and can be switched on or off at any time, which predetermines the random nature of the network structure. In such networks, the nodes are fully or partially functionally identical. The peer-to-peer principle of organizing dynamic networks determines their high fault tolerance by eliminating the vulnerability of the central link, which is characteristic for systems with asymmetric functionality. In the case of communication networks, gateways or routers can represent such a link. Since each node is required to act as a router in the ad hoc network, the failure of any one of them is not critical for the network as a whole. In addition, the symmetry of the functionality of the nodes of the network creates the prerequisites for giving it the property of self-organization, which makes the network not only fault-tolerant, but also scalable and scalable [16].

Thus, the main properties of ad hoc networks are that they:

- wireless;
- dynamic;
- decentralized.

The decentralization of the network is the absence of a single management center [19].

A distinctive feature of the above technologies from other dynamic networks is the spatial stationarity of the nodes of the network. This greatly simplifies the solution of the task of routing data flows, since the dynamics of the network structure is manifested only in the fact that nodes can leave the network, which leads to the termination of the routes passing through them. Stativity of network nodes determines the number of neighbors that is limited for a given node and, thus, allows to create and store a complete network topology or its separate fragments on network nodes [16].

C. MANET networks

MANET (Mobile Ad hoc NETworks) networks are radio networks with random mobile subscribers, implementing fully decentralized control in the absence of base stations or reference nodes. The topology of such networks is rapidly changing with random connection of nodes [18].

The MANET network is a special case of an ad hoc network.

The main properties of MANET networks are that they:

- wireless;
- dynamic;
- decentralized;
- mobile.

The mobility of networks is the ability to move network nodes in space [19].

D. VANET networks

VANET (Vehicle Ad hoc NETworks) are vehicle communication networks. They are hybrids of MANET networks [18].

A distinctive feature of such networks is that all nodes are constantly moving and can communicate with each other for a very short time, which greatly complicates the information routing.

In some systems, the end devices can be constantly moved and should be provided with communication at any point. In this article, we will look at the device authentication protocol, which, on the contrary, implies the static nature of the subscribers, and ensures that the device cannot work at its considerable distance from the place of its initialization.

III. Protocols of authentication of devices in sensor networks

Over the past few decades, many protocols have been proposed for authenticating devices in sensor networks. Most of them are based on the ideas of several basic protocols, which we briefly describe below.

A. EG Scheme

In the scheme proposed by Esheneauer and Gligor [20], before deployment, the server must generate a large pool of keys and write to each device a randomly selected subset from this pool. After deployment, any two neighboring network devices with a certain probability will have at least one shared key.

This basic scheme has a number of shortcomings:

- The device ID is not used, which does not allow to find out which device is being authenticated.
- When capturing one device, all devices that have at least one public key with this device are compromised.
- When scaling the network, large subsets of keys must be written to the devices, otherwise, the probability of coincidence of at least one key on two neighboring devices will be very small.

Later this scheme was improved. In RKS-K [21] devices for authentication need not one, but several common keys, on the basis of which the pair key is considered. Also in this scheme, it is proposed to use identifiers in addition to common keys, which makes it possible to uniquely identify the interlocutor. To update the pairing key, it is suggested to transfer it by parts on several different routes, which will not allow the attacker to intercept the new key if he does not have access to a sufficient number of neighboring devices. Despite the increase in the cryptographic strength of such a scheme, it retains a number of shortcomings:

- A large communication resource is used.
- With the increase in the number of captured devices, the probability of compromising the entire network increases.
- The constraint on a maximum number of devices in the network is retained.

In PKS-MP scheme [22] was suggested to write a subset of keys to devices based on the probability of where the device will be installed. If two devices are most likely to be installed side by side, the same shared key will be added to their subsets. Thus, fewer keys could be written to the devices, which removes the restriction on the size of the network. However, in practice, it is rather difficult to determine in advance where a particular device will be installed.

B. TESLA

A digital signature requires asynchrony. Nevertheless, classical asynchronous algorithms require a large computing resource. In [23] it was suggested to achieve asynchrony with time by using only synchronous cryptographic primitives. In this scheme, the message and signature are sent at different time intervals. The signature is based on a keychain, where everyone can calculate any previous key, but the next key can only calculate a device that has a secret.

In this scheme:

- For the operation of the circuit, it is necessary to synchronize the devices in time.
- There is a big delay in calculating the key.
- For large networks, a very large number of messages are required.
- Each device is activated separately during the initial initialization.

Based on this scheme were μTESLA [24] and Multilevel μTESLA [25] proposed, optimized for large networks by the use of broadcast.

Later the RPT scheme [26] was proposed, which allows instant authentication of the message, but requires sending messages at regular and predictable intervals.

C. BiBa

In BiBa [27] to sign the message, the sender first computes the hash from the message and selects on its basis one one-way function from a certainly predetermined subset of one-way functions. With its help, the sender also considers hashes from pre-generated random numbers and seeks a collision among the resulting values. The collisions found will be the signature of the message. The cryptographic stability of this protocol is ensured by the fact that the sender will find collisions with greater probability that an attacker who does not know the entire set of random numbers generated by the sender even if he has captured several devices.

In this scheme:

- Messages are quickly verified.
- Messages signs for a very long time.

In HORS [28] protocol allows you to sign messages faster because the choice of a one-way function from a predetermined subset is optimized.

D. LEAP

Basic LEAP [29] protocol is designed for use in a hierarchical mesh network. In this case, several types of keys are allocated for different types of messages. The private key is used to encrypt messages between the device and the server. The device's paired key is generated with its neighbors based on the master key (MK) and device IDs. The master key is deleted from the device memory after the pairing keys are created. The cluster key is generated by one of the devices for all neighboring keys and is transmitted to them by means of paired keys. The group key is one for all network. One of the drawbacks of this scheme is that if an attacker manages to get the master key from the device's memory before it is deleted, then it gets access to the entire network.

In TB-LEAP [30] for each time interval, you use your master key, which narrows the area of compromised devices when you capture the master key from the entire network to one cluster. Next, we will look more closely at the LEAP protocol, and suggest an improvement of the scheme considered in [31].

TABLE I. COMPARISON OF PROTOCOLS OF AUTHENTICATION OF DEVICES IN SENSOR NETWORKS

Protocol	**Property**				
	Resistance to device compromise	Immediate authentication	Message sent in irregular times	Determinism	Scalability
μTESLA	+	-	+	+	-
Multicast μTESLA	+	-	+	+	+
RPT	+	+	-	+	+
EG	-	+	+	-	-
PKS-K	-	+	+	-	-
PKS-MP	-	+	+	-	+
BiBa	+	+	+	-	+
LEAP	+(-*)	+	+	+	+
TB-LEAP	+(-*)	+	+	+	+

In Table I, the described protocols are compared at a qualitative level according to the criteria proposed in [26].

Resistance to device compromise is one of the most important properties of the authentication protocol in the sensor network. It implies that the device captured by an attacker does not allow to compromise the entire network.

Immediate authentication means no authentication delay, which occurs in protocols based on TESLA.

Some protocols, for example, RPT, require sending messages at regular intervals, which can be inconvenient or even unattainable in practice.

The scalability feature includes the ability to connect a large number of devices, as well as the ability to connect new devices after the initial deployment phase is over.

Many authentication protocols are probabilistic. In them, cryptographic material is randomly distributed across devices, and there is a possibility that neighboring devices cannot generate a paired key or they will need a large number of attempts.

As can be seen from Table I, among the deterministic protocols, three groups can be distinguished according to the existing flaws (- in the table). When analyzing the LEAP protocols in this paper, a vulnerability at the initialization

stage (which is marked in the table with the sign -*) was detected.

IV. LEAP VULNERABILITY TO ATTACK AT DEVICE INITIALIZATION PHASE

In basic LEAP protocol during device initialization, the device sends broadcast requests with its own identifier and waits for a response from neighboring devices with their identifiers. All identifiers are not encrypted.

$u \rightarrow *: ID_u$

$v \rightarrow u: ID_u, H(H(ID_v||MK), ID_u||ID_v)$

where u – is a new device; v – one of the devices that responded to the request of u; $H()$ – cryptographic hash function.

In such a scheme, a vulnerability is possible, which can be used to compromise the entire network by capturing just one device. Next, we consider in more detail the version of the attack and ways to protect it from it.

Algorithm 1 Attack on LEAP

1: Device u during initialization sends a broadcast request:
$$u \rightarrow v: ID_u.$$
2: Device v responds:
$$v \rightarrow u: ID_u, H(H(ID_v||MK)||ID_u||ID_v) \quad (1)$$
where ID_u is u identifier,
ID_v is v identifier,
MK – is master key.
3: Attacker E intercepts (1).
4: Device u generates a pairwise key:
$$K_{u,v} = H(ID_u||ID_v||MK),\ ID_u < ID_v$$
5: A new device k is added to the network and broadcasts:
$$k \rightarrow *: ID_k$$
6: Attacker E responds with a response intercepted previously:
$$E \rightarrow k: IDu, H(H(IDv)||\ IDu||IDv)$$
7: Device k based on the response of E calculates pairwise key:
$$K_{k,v} = H(ID_k||ID_v||MK),\ (ID_k<ID_v)$$
8: Attacker captures k, and gets the key $K_{k,v}$.
9: An attacker can communicate with v appearing as k.

Thus, an attacker can get paired keys to communicate with the devices he needs v by intercepting their responses (1) to the initialization request and capturing one device k. Next, we will propose and analyze the possible options for protection against the described attack.

V. POSSIBLE SUPPLEMENT OF THE LEAP PROTOCOL TO PREVENT VULNERABILITY AT THE INITIALIZATION STAGE

Depending on the specific implementation of the system, we can use devices that expend different energy resources for different operations. Next, we will consider two schemes, and when choosing the optimal of these two schemes, you should consider which of the operations in a particular system requires less resource: wireless data transfer or digital signature verification.

A. Symmetrical scheme:

The scheme includes the phase of setting up devices on the server, deploying the network and the phase of adding new devices.

When initialized, the server generates a unique identifier for each device, the master key and writes them to the devices. Next is the phase of deployment of the network, during which all devices are new i.e. they still store the master key.

Algorithm 2 Symmetric deployment

1: Device u broadcasts:
$$u \rightarrow *: ID_u, H(ID_u||R_u||MK), R_u, new$$
where R_u – this is a random number generated by u,
new – is a string that allows you to understand that the device is still storing MK.
2: Device v calculates $H(ID_u||R_u||MK)$, and
if it coincides with what u sent, **then** sends:
$$v \rightarrow u: ID_v, H(ID_v||R_v||MK), R_v, new$$
where R_v – this is a random number generated by v.
3: Device u verifies signature, and
if it is correct **then** calculates
$$K_{u,v} = H(ID_u||ID_v||MK),\ ID_u < ID_v \quad (2)$$
4: Device v calculates
$$K_{v,u} = H(ID_u||ID_v||MK),\ ID_u < ID_v \quad (3)$$
5: Device u generates a key
$$K_{u,u} = H(ID_u||MK) \quad (4)$$
needed for further network scaling.
6: Device v generates key $K_{v,v}$ similarly to (4).
7: Device u deletes MK.
8: Device v deletes MK.

As we can see (2) and (3) have the same values, which will then be used as a pair key for u and v devices. During symmetric initialization of the network, an attacker can intercept device identifiers and random numbers generated by them, but without the MK, this will not allow him to calculate the paired key.

Since after the initial deployment of the network, the installed devices delete the master key, we need a different scheme to add new devices to the network.

Algorithm 3 Symmetric new device addition

1: New device u broadcasts:
$$u \rightarrow *: ID_u, H(ID_u||R_u||MK), R_u, \text{new} \quad (5)$$
where R_u – this is a random number generated by u.
2: Device v responds with its identifier with a confirmation ACK_v:
$$v \rightarrow u: ID_v, ACK_v = H(ID_u||ID_v||R_u||K_{v,v}), old \quad (6)$$
where old – this is a string informing that v no longer stores MK.
3: Device u calculates $K_{v,v}$ according to (4), and
if calculated value $H(ID_u||ID_v||R_u||K_{v,v})$ coincides with ACK_v **then** saves $K_{v,v}$
else deletes $K_{v,v}$.
4: Device u deletes MK.

It can be seen that an attacker can not, intercepting the request for initialization (5) of the device u, use this request

elsewhere in the network to force the new device v to generate the pair key K_v, u because the device u the attacker can not sign the random number R_u, and u will delete the key. But in this case, the device *v* must respond to each request (5), and in return must consider and forward (6).

B. Asymmetrical scheme:

Algorithm 4 Asymmetric pre-deployment

1: Server generates keys for digital signature *(MK = K_{priv}, K_{pub})* and writes them to devices. In this case, the master key in such a scheme is the secret key K_{priv}.
2: Server generates a unique identifier for each device, and writes them to devices.

Algorithm 5 Asymmetric deployment

1: Device *u* sends signed:

$$u \rightarrow *: ID_u, H_{MK}(ID_u)$$

where $H_{MK}()$ — is a signature with the *MK*.
2: Device *v* verifies the signature with a public key K_{pub}, and
if the signature is correct, **then** sends:

$$v \rightarrow u: ID_v, H(ID_v||R_v||MK), R_v, new$$

3: Device *u* verifies the signature and
if it is correct **then** calculates $K_{u,v}$ similarly to (2).
4: Device *v* calculates $K_{v,u}$ similarly to (3).
5: Device *u* generates key $K_{u,u}$ similarly to (4).
6: Device *v* generates key $K_{v,v}$ similarly to (4).
7: Device *u* deletes *MK*.
8: Device *v* deletes *MK*.

Algorithm 6 Asymmetric new device addition

1: Device *u* sends:

$$u \rightarrow *: ID_u, R_u, H_{MK}(ID_u||R_u)$$

2: Device *v* verifies signature and **if** it is correct **then** responses:

$$v \rightarrow u: ID_v, ACK_v = H(ID_u||ID_v||R_u||K_{v,v}), old$$

3: Device *u* calculates $K_{v,v}$ similarly to (4) and
if $H(ID_u||ID_v||R_u||K_{v,v})$ coincides with ACK_v, **then** saves $K_{v,v}$
else deletes $K_{v,v}$.
4: Device *u* deletes *MK*.

The asymmetric scheme differs from the symmetric one in that the first broadcast request is also signed.

It is important to note that the performance of an asymmetric operation will use much more energy. However, its use will make it possible while a DoS (Denial of service) attack not to respond to an attacker's messages. Thus, if there are no attacks during the network operation, the devices will perform only one extra asymmetric operation, and in the attack they will not spend the resource on the answers, but they will have to consider asymmetric operations.

We also note that either in the symmetric scheme or in the asymmetric at the stage of deployment and initialization of the new device, the new devices store the master key, so the capture of the device during this phase will allow an attacker to compromise the entire network.

VI. DEVICE AUTHENTICATION

When choosing an authentication protocol for an ad hoc network, it is important to understand that in addition to threats such as unauthorized access and data substitution, there is also a threat of battery discharge of the device [9]. In order to counter this threat, it is necessary to reduce the computing costs of the device during authentication.

Considered protocol:

Algorithm 7 Device authentication

1: Device *u* sends authentication request:

$$u \rightarrow v: ID_u \quad (7)$$

where *u* – the device that wants to send a message, *v* – receiving device.
2: Device *v* sends random number *Rv* with an acknowledgment:

$$v \rightarrow u: R_v$$

3: Device *u* calculates the session key:

$$K_{auth} = H(K_{u,v}||ID_u||R_v) \quad (8)$$

4: Device *v* calculates the session key similar to (8).

As seen from (7), authentication is always initiated by the sending device. The role of the receiving device is to generate and transmit a random number for each device that wants to authenticate, and calculate the session key based on it and passively listen to the channel.

It is important to note that due to the use of a cryptographic hash function, for example, [32], it is not necessary to change the session key when the device attempt fails to authenticate. The new key will be generated only for the new session.

VII. ATTACKS TO SENSOR NETWORKS

Since there can be many devices in the sensor network and they can be spread over a large area, often an attacker can gain direct access to the device [23]. Therefore, it makes sense to consider separately those threats in which an attacker has access to the device, and those for which only wireless access is used.

A. Attacks without access to the device:

Being in the WSN radio zone, an attacker can intercept traffic and create malicious traffic.

Here, the following attacks can be carried out:

1) *DoS*

DoS attack is an attempt to make the device inaccessible to its real subscribers. Due to the very limited computing resources and the battery charge, IoT devices are particularly vulnerable to this type of attack.

The proposed protocol mitigates the impact of such attacks because it frees the device from having to calculate a new session key every time another device attempts to authenticate.

When the device receives a request for an attempt to authenticate it must count (8). In this case, if the transmitting device successfully passes authentication, then the receiving device exits the energy-saving mode and continues to service the trusted subscriber. If the transmitting device does not authenticate, the receiving device does not recalculate the session key and does not get out of sleep.

An attempt may also be made to launch a DoS attack not with messages, but with authentication requests. To avoid such an attack, two schemes are proposed in this paper. The synchronous scheme uses only symmetric cryptographic primitives to verify the authenticity of the interlocutor but requires one additional transfer. The asymmetric scheme allows not to respond to non-trusted requests, but it uses asymmetric primitives that consumes more energy.

2) *Interception*

Listening to the channel may allow an attacker to obtain confidential data. To avoid this, the transmitted messages must be encrypted. To simplify key management during encryption, in some cases, you can use the session key (8). However, we recommend using a unique key to encrypt the transmitted data on each device and decrypt them not on every intermediate node of the network, but on the server. In this case, the keys for decryption must be stored on the server.

3) *Spoofing*

To protect data from spoofing, it is not enough to encrypt them. You have to use a digital signature for the messages you send. For this procedure, it is also possible to use a session key (5).

B. Attacks with access to the device:

After accessing the device, an attacker can access the device's memory.

Here the following attacks can be carried out:

1) *Spoofing*

If an attacker has access to the device's memory, he can extract cryptographic material and partially or completely replace the device [14].

At the same time, it is important not to allow an attacker who gained control over one device to gain control over the entire sensor network. This requirement is ensured by the fact that for each pair of devices the session key is unique. If the device is added to the network after initial network deployment, the paired key (4) may not be unique, but by using the transmitter's ID in (8), the session key will be unique. Thus, an attacker can only forge an authentication procedure from a captured device.

It is believed here that the MK was removed from the device before the attacker got access to it.

2) *Interception*

By controlling the device, an attacker can scan traffic passing through it. To avoid scanning telemetry data on the repeater, the data could be decrypted not on each node, but on the server.

3) *Moving the device*

When an intruder accesses the device, he can move it to a location from which the device will send incorrect telemetry data or even clone the device.

The proposed authentication protocol allows you to move the device only within the scope of the device with which it is authenticated. If you move the device from another part of the network, it will not be able to generate pair keys with new neighbors because for this MK is needed. Thus, illegal movement of the device is impossible.

VIII. THE USE OF CONSIDERED PROTOCOL

The proposed LEAP-IP protocol is designed for use in ad hoc networks since it is designed for a point-to-point interaction of nodes with symmetric functionality. However, it does not prevent to use it in mesh networks with a complex hierarchy.

Despite the fact that the protocol is not suitable for use in MANET and VANET networks, this does not mean that its subscribers cannot move. There are possible system architecture options in which mobile subscribers collect information for some time, and transmit it to the repeater by returning to the initialization point of the device.

In addition to authentication, the protocol can provide keys that can be used to encrypt and sign the transmitted messages.

Since the algorithm guarantees the inability to use the device outside its initialization zone, it can be used for local positioning. Such a system will not be very accurate but does not require any additional equipment of the device.

Depending on the characteristics of the particular system, the use of an asymmetric scheme will be optimal if the device spends more resources on the data transfer than on the digital signature calculation and the use of symmetrical scheme otherwise.

Since the master key is used for the calculation of the paired keys during the initialization phase of the network, it is important to prevent the key from being received by the attacker. After receiving the MK attacker can forge messages from any device on the network.

It is important to understand that to scale the network you will have to store the MK on the server to record its not new devices. Therefore, it is necessary to protect not only new devices from physical attacks before they are initialized, but also to protect the MK on the server.

We also cannot unquestioningly trust the data that came from the sensor network. It needs to be checked and logged on server.

IX. CONCLUSION

In this paper, a simple and effective LEAP-IP protocol for authenticating devices in a sensor network is proposed, which ensures that the device cannot be moved unauthorized in a wireless sensor network. The proposed protocol provides protection from interception and spoofing data at all phases of

the network operation, and also avoids large energy losses when the system is attacked.

This protocol is suitable for use in non-mobile ad hoc and mesh networks and provides simple and secure scalability of the system.

In the future, it is supposed to simulate the operation of the protocol, determine the optimal size of the master key and device identifiers, and also select an energy efficient hash function for use in the proposed protocol.

ACKNOWLEDGMENT

The work of the second author is supported by the Academy of Finland.

REFERENCES

[1] A. Safonov, “In IoT Things generate information themselves”, *PostNauka*, August 2014. Web: https://postnauka.ru/talks/30032.

[2] E. Horov, “From sensor networks to Internet of Things”, *PostNauka*, October 2017. Web: https://postnauka.ru/faq/80050.

[3] L. Voskov, “IoT evolution”, *PostNauka*, September 2017. Web: https://postnauka.ru/talks/80081.

[4] SAP Cloud Platform official blog, “Step by step: collect and test the Internet of things based on the SAP Cloud Platform”, April 2017. Web: https://habrahabr.ru/company/sap/blog/326526/.

[5] Business Wire Inc., “DC Forecasts Worldwide Spending on the Internet of Things”, December 2017. Web: https://www.businesswire.com/news/home/20171207005963/en/IDC-Forecasts-Worldwide-Spending-Internet-Things-Reach.

[6] B. Buntz, “The 10 Most Vulnerable IoT Security Targets” in Internet of Things Institute, July 2016. Web: http://www.ioti.com/security/10-most-vulnerable-iot-security-targets.

[7] B. Buntz, “25 leading IoT security companies”, *Internet of Things Institute*, July 2016. Web: http://www.ioti.com/security/25-leading-iot-security-companies.

[8] C. Jamie, “How to Get One Trillion Devices Online” in MIT Technology Review, September 2017. Web: https://www.technologyreview.com/s/608878/how-to-get-one-trillion-devices-online/.

[9] ZingBox Inc., “What Makes IoT Security so Unique”, January 2018. Web: https://www.zingbox.com/iot-security/.

[10] D. V. Ragozin, “Synchronized sensor networks modeling”, Programming problems, vol. 2-3, 2008, pp. 721-729.

[11] Strizh Telematica, “From innovative LPWAN technology to the first IoT platform”, Embedded day, 2016. Web: http://www.embeddedday.ru/2016/presentations/1.5_%D0%A1%D0%A2%D0%A0%D0%98%D0%96_LPWAN.pdf.

[12] CENTRI Technology Inc., “CENTRI Internet of Things Advanced Security”, January 2018. Web: https://www.centritechnology.com/wp-content/documents/CENTRI_datasheet_IoTAS.pdf.

[13] T. Story, “Going back to school on IoT security – personal reflections from a cybersecurity product marketeer”, September 2016. Web: https://blogs.cisco.com/security/going-back-to-school-on-iot-security-personal-reflections-from-a-cybersecurity-product-marketeer.

[14] Microsoft Inc., “Internet of Things security architecture”, January 2018. Web: https://docs.microsoft.com/en-us/azure/iot-suite/iot-security-architecture.

[15] Google Cloud Platform, “Device Security”, February 2018. Web: https://cloud.google.com/iot/docs/concepts/device-security.

[16] M. G. Shishaev, “Modern technologies of ad-hoc type networks and possible approaches to the organization of peer-to-peer telecommunications networks based on mobile devices of short range”, *Proceedings of the Kola Science Center of the Russian Academy of Sciences*, Kola Science Center 2010, pp. 70-74.

[17] Y. Rohilla, “A comparative study of wireless mesh and ad hoc”, *International Journal on Computer Science and Engineering (IJCSE)*, vol. 4, № No. 06, June 2012. pp. 1181-1184.

[18] J. Voitenko, “What is MANET or why WiFi is not the solution to all telecommunication problems”, Web: https://habrahabr.ru/post/197860/.

[19] S. V. Guss, “Self-organizing mesh networks for private use”, *Mathematical structures and modeling*, vol. 4(40), 2016, pp. 102-115.

[20] L. Eschenauer, V. D. Gligor, “A Key-management Scheme for Distributed Sensor Networks,” *in Proc. of the 9th ACM Conference on Computer and Communication Security (CCS’02)*, pp. 41-47, Nov. 2002.

[21] H. Chan, A. Perrig, D. Song, “Random Key Predistribution Schemes for Sensor Networks,” *in IEEE Symposium on Research in Security and Privacy*, pp. 197-213, 2003.

[22] W. Du, J. Deng, Y. S. Han, P. K. Varshney, “A Key Predistribution Scheme for Sensor Networks Using Deployment Knowledge,” *in IEEE Transactions on dependable and secure computing*, vol. 3, no. 1, pp. 62-77, Feb. 2006.

[23] A. Perrig, R. Canetti, D. Song, D. Tygar, “Efficient authentication and signing of multicast streams over lossy channels”, *in Proceedings of the 2000 IEEE Symposium on Security and Privacy*, 2001.

[24] A. Perrig, R. Szewczyk, V. Wen, D. Culler, D. Tygar, “SPINS: Security protocols for sensor networks” *in Proceedings of Seventh Annual International Conference on Mobile Computing and Networks*, 2001.

[25] L. Donggang, N. Peng, “Multilevel μTESLA: Broadcast Authentication for Distributed Sensor Networks”, North Carolina State University, *ACM Transactions on Embedded Computing Systems*, Vol. 3, No. 4, November 2004, pp. 800–836.

[26] M. Luk, A. Perrig, B. Whillock, “Seven Cardinal Properties of Sensor Network Broadcast Authentication”, Electrical and Computer Engineering Carnegie Mellon University, pp. 147-156.

[27] A. Perrig, “The BiBa one-time signature and broadcast authentication protocol”, Eighth ACM Conference on Computer and Communication Security, 2001, pp. 28–37.

[28] L. Reyzin, N. Reyzin, “Better than BiBa: Short One-time Signatures with Fast Signing and Verifying”, Boston University, pp. 1-9.

[29] S. Zhu, S. Setia, S. Jajodia, “LEAP: Efficient security mechanisms for largescale distributed sensor networks,” *in Proceedings. of the 10th ACM Conference on Computer and Communication Security (CCS’03)*, November 2003 pp. 62-72.

[30] J. Jang, T. Kwon, J. Song, “A time-based key management protocol for wireless sensor networks”, *Proceedings of ISPEC*, 2007, LNCS 4464, pp. 314-328.

[31] S. V. Nesteruk, A. V. Schiscko, S. V. Bezzateev, “Aspects of wireless sensor network security”, *International SUAI student conference*, vol. 1 Engineering science, 2017, pp. 282-285.

[32] I. Anshel, D. Atkins, D. Goldfeld, P. Gunnells, “A class of hash functions based on the algebraic eraser”, *Groups Complex*, Cryptol, 2016, vol:8(1), pp. 1–7.

[33] M. Abomhara, G. M. Koien, “Cyber Security and the Internet of Things: Vulnerabilities, Threats, Intruders and Attacks”, *Journal of Cyber Security and Mobility*, University of Agder, May 2015, pp. 65-88.