Position papers of the 2014 Federated Conference on Computer Science and Information Systems pp. 183–190
DOI: 10.15439/2014F176
ACSIS, Vol. 3

The concept of authentication in WSNs using TPM

Janusz Furtak
Military University of Technology
ul. Kaliskiego 2,
00-908 Warszawa, Poland
Email: jfurtak@wat.edu.pl

Jan Chudzikiewicz
Military University of Technology
ul. Kaliskiego 2,
00-908 Warszawa, Poland
Email: jchudzikiewicz@wat.edu.pl

Abstract— This document describes how to use the Trusted Platform Module (TPM) to authenticate sensors in wireless sensors network which create a sensors' domain. Model of the wireless sensor network is presented. There are three types of nodes in the domain. The M node is an authentication authority in sensors' domain – it stores credentials of all nodes of domain. The M node is also the recipient of the data emitted by the domain sensors. The S node is the source of sensors data (i.e. air temperature, concentration of sulfur dioxide, wear of ammunition, etc.). The rM node is acting as backup for M node. The concept of main operations available in the sensors' domain related to: managing of sensors in the domain, authentication of sensors and regeneration of the node credentials is presented. The concept is a proprietary solution developed by the authors of the paper.

I. Introduction

The wireless sensor networks (WSNs) consist of large number of ultra-small, low-power and inexpensive wireless sensor nodes with sensing, computing and communication capabilities [1], [2]. The popularity of the WSNs causes that are used in many areas like for example: military, ecological, health-related areas etc. These applications often include the monitoring and processing of sensitive information or location of soldiers on the battlefield. Security is therefore important in WSNs. We need use secure communication mechanisms in WSN to ensure confidentiality, authenticity and integrity of the nodes and data. Security mechanisms deployed in WSNs should involve collaborations among the nodes due to the decentralized nature of the networks and absence of any infrastructure. The situation becomes critical when the nodes are equipped with cryptographic materials such as keys and other important data in the sensor nodes. Moreover, adversaries can introduce fake nodes similar to the nodes available in the network which further leave the sensor nodes as un-trusted entities.

The researchers in WSN security have proposed various security schemes which are optimized for these networks with resource constraints. A number of secure and efficient routing protocols [3], [4], secure data aggregation protocols [5], [6], [7], [8] and additional security mechanisms such as Trusted Platform Module (TPM) [9], [10], [11], [12] etc. has been proposed by several researchers in WSN security. Taking this into consideration WSNs among others could be divided into different security levels [13], [14]:

- **Availability**, which ensures that the desired network services are available even in the presence of denial-of-service attacks;
- **Authorization**, which ensures that only authorized sensors can be involved in providing information to network services;
- **Authentication**, which ensures that the communication from one node to another node is genuine, that is, a malicious node cannot masquerade as a trusted network node;
- **Confidentiality**, which ensures that a given message cannot be understood by anyone other than the desired recipients
- **Integrity**, which ensures that a message sent from one node to another is not modified by malicious intermediate nodes;
- **Nonrepudiation**, which denotes that a node cannot deny sending a message it has previously sent;
- **Freshness**, which implies that the data is recent and ensures that no adversary can replay old messages.

Most of them are very important for military applications. Secrecy is part of its nature; and data (sensed/ aggregation/processing) is required to remain confidential. This is critical to the successful operation of a military application. Enemy tracking and targeting are among the most useful applications of wireless sensor networking in military terms. Considering the above, the secure method of transmitting and storing data in WSNs is proposed in the paper. The Trusted Platform Module (TPM) is the basis of the presented method. A TPM is used for secure storing the necessary data to authenticate the nodes, and generate symmetric keys, and asymmetric keys (private/public).

The second section provides proposed architecture of WSNs, and basic definitions. The basic data stored in every nodes (depending on the role they played in the network e.g. domain master (node M), and slave (node S)), and the basic data structures used in the nodes are defined in the section. In the third section the procedures to ensure proper authentication of sensors in domain and

correct data transfer between sensors are described. Finally, a few concluding remarks are presented.

This concept is a proprietary solution developed by the authors of the paper. Some inspirations for the development of the presented concept the authors of the method drew from solutions used in DNS.

II. The Model of Wireless Sensor Network with Authentication

In the domain of sensors there are two authorities. The first is the node (Data Collector) which is the recipient of the data emitted by the domain sensors. The node which manages the Root of trust is the second authority. The Root of trust is to be used to authenticate all sensors involved in the exchange of data between elements of the domain of sensors. The second authority is to act as a master of domain and will be called the node M. The presented concept assumes that both the role of the recipient of data from the sensors (i.e. Data Collector) and the role of the master of domain plays the same node.

In the sensors' domain is exactly the one node that acts as the domain master (node M). To this domain belong sensors of type slave (nodes S), which are registered by the node M. Nodes S are the source of data. Node S is initiated and authenticated by node M of domain. Node M stores the root trust of sensors' domain. The sensors' domain structure is shown in **Fig. 1**

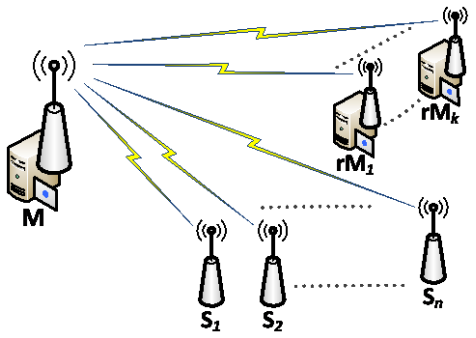

Fig. 1 The structure of sensors' domain

In the domain may be designated nodes acting as backups masters (replicas of master - rM). Such a node may be a S node after the establishment the role rM for him, on condition that its hardware and software resources provide this capability. In the domain may be no node type rM (this is not recommended), but there may be a few such nodes. The task of node rM is to store a copy of the root of trust from the node M of domain. Updating the copy should be done by one method selected from the following [1]:

- after the modification of the root of trust on the node M;
- at fixed intervals of time;
- on demand of node M;
- on demand of node rM.

[1] Analysis of the advantages and disadvantages and the choice of how to upgrade the resources of nodes rM are not subject of this study.

From the viewpoint of authentication procedures nodes M and rM for nodes S are the same. Node rM can become a new node M of domain after changing its role, due to proven inactivity of old node M. In this case the node, which has so far acted as a node M, becomes a node rM, or node S, or is removed.

When the sensor does not function, is turned off or damaged, it is assumed that this node is in a non-active state, and when the sensor is functioning, then the node is in the active state.

If a node acts as M and remains in a non-active state for longer than a predetermined period of time, the procedure for the designation of the "new" master of the nodes is started. The "new" master is designated from among nodes, which until now were playing the role of rM. When in the domain there are no nodes rM, the new master shall be designated among the S nodes. A node that has lost the role M as a result of prolonged inactivity, but retains the efficiency, after obtaining the active state can act as rM or only S (it might be needed restart the procedure for initiating node).

If a node acts as rM and remains in a non-active state for longer than a predetermined period of time, and there are no others nodes rM in the domain, procedure for designating "new" node rM from nodes S is started. A node that has lost the role rM as a result of prolonged inactivity, but retains the efficiency, after obtaining the active state can act as rM or only S (it might be needed restart the procedure for initiating node).

Sensor, which acts as a node M receives data from S nodes.

Minimum requirements for a sensor type S are as follows:

- sensor must be equipped with a TPM (see the next section);
- sensor must have an interface that allows direct connection to the node M (e.g. via USB) in the registration procedure of the node in the domain;
- the ability to send sensor data (i.e. measurement data) to M node using only wireless connection.

In order to enable automatic authentication procedure of the node and regeneration procedure for S node credentials, S node should be able to receive data transmitted by node M via a wireless connection. Otherwise, the node authentication procedure is not possible and change of credentials of this node will be possible only after the re-registration of the node. Nodes that are designed to play the role of M or rM must be able to bi-directional communication with other nodes, and should also have adequate resources in terms of power, processing capability and storage capacity.

A. Trusted Platform Module

In the presented model for authentication sensors are used mechanisms offered by the Trusted Platform Module (TPM). It is assumed that each element of the domain of sensors is equipped with TPM.

TPM is an implementation of a standard developed by the Trusted Computing Group [15]. This module is designed to support the cryptographic procedures and protocols that can be used for securing data [16]. Trusted Platform Module provides the following functions:

- generating an asymmetric key pair,
- secure storage of keys,
- generating an electronic signatures,
- encryption and decryption,
- implementation of an operation defined by the standard PKCS #11.

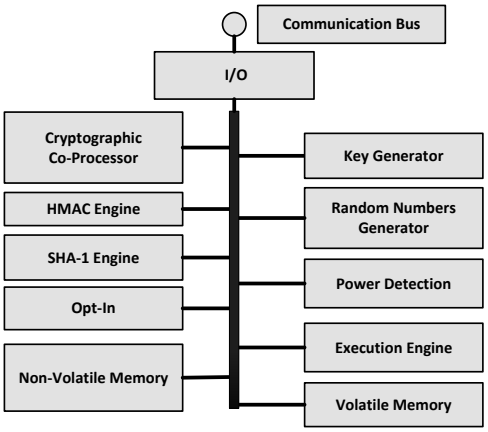

Fig. 2 TPM Component Architecture (based on [15])

The following algorithms are typically implemented in TPM [17]: RSA, SHA-1, HMAC and AES[2]. In addition, each TPM chip stores a unique serial number and its RSA private key that is never available to read. TPM components are shown in **Fig. 2**.

B. Resources of sensors

Each sensor is equipped with a TPM. In the resources of TPM are stored the necessary data to authenticate the node acting as the S in domain. The structure of the data is shown on **Fig. 3**. Sensors, which are to play the role of M or RM must be equipped with additional memory, which is intended to store the description of the domain and descriptions of remaining domain nodes.

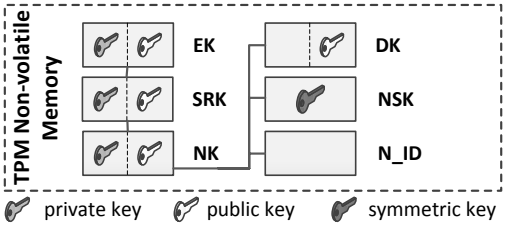

Fig. 3 The data stored on S node

Content of credentials stored in non-volatile memory of the TPM, which are used by a node S (Slave data):

- EK (Endorsment Key) - key pair (private/public) generated in the production phase of the TPM;
- SRK (Storage Root Key) - key pair (private/public) generated during the process of initiating the TPM in the procedure for registering a S node in the domain of sensors;
- NK (Node Key) - key pair (private/public) of node; generated during the procedure for registering a S node in the domain of sensors; acts as the parent for the remaining keys stored in the resources of the node;
- DK (Domain Key) – public part of the key of sensors' domain to which the node belongs; obtained during the procedure for registering the node in the domain;
- NSK (Node Symmetric Key) – symmetric key to encrypt the data sent from this node to M node; obtained during the procedure for registering the node in the domain and renovated in the regeneration procedure of S node credentials;
- N_ID (Node ID) – ID of the sensor (e.g. IPv6 link local address of sensor).

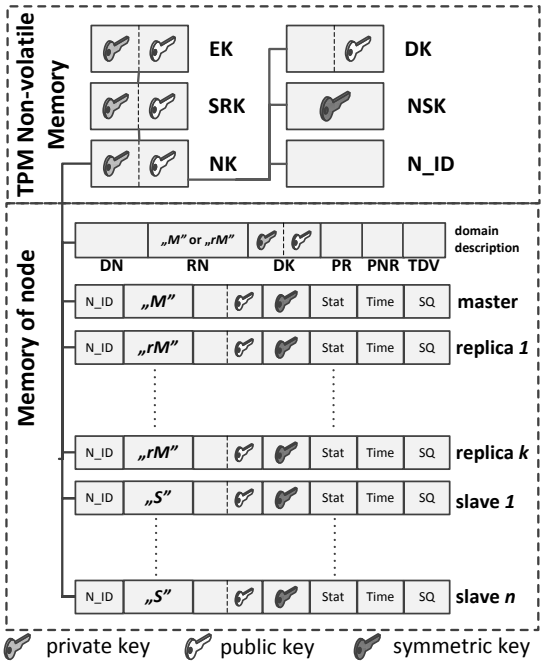

Fig. 4 The data stored on M node or rM node

Credentials stored by the node M (the structure of the data is shown on **Fig. 3Fig. 4**) consist of three resources: Slave Data, Domain description and Description of domain nodes. The content of these resources is as the following:

- **Slave data** (the same as for the S node);
- **Domain description**:
  - DN (Domain Name) – the name of domain;
  - RN (Role of Node) – determine whether below data are the resource of master node or the resource of replica of master; it is synonymous with the role it plays in the domain; may have one of values: M or rM;
  - DK (Domain Key) – key pair (private/public) of sensors' domain; generated during the procedure generated in the process of creating the domain

[2] TPM uses a symmetric algorithm AES to protect the confidentiality of the session in which it participates following the recommendations of the TCG. However, symmetric encryption functions are not normally accessible outside the TPM.

of sensors and establishing the role of the "master" in the domain for the first node;

- PR (Period of Replication) – the time after which the rM node is required to establish communication with the node M and refresh the domain data;
- PNR (Period of Non-success Replication) - the time after which the node rM is obliged to repeat the attempt to establish communication with the M node if the previous attempt refreshing the domain data was not successful;
- TDV (Time of data validity) – after this time and the inability to refresh, the domain data are invalid and node becomes a node S.

Fig. 5 The data structure describing a node

- **Description of domain nodes**. Description of each node contains the following data (the structure of the data is showed on Fig. 5):
  - N_ID (Node ID) – ID of the sensor;
  - RN (Role of Node) – the role filled by the node in the domain; it can take values from the set {M, rM, S};
  - SlvK - public part of an asymmetric key N_ID node of sensors' domain;
  - NSK - symmetric key to encrypt the data sent from this node to M node; obtained during the procedure for registering the node in the domain and renovated in the procedure for the regeneration of S node credentials;
  - Stat - status of the node; it can take one of the values: non-active(-1), active(0), active non-confirmed (n), where n is the number of consecutive unsuccessful attempts to establish communication with the node
  - Time - moment of the last and the effective transmission [3];
  - SQ - the sequence number of the last sent frame (modified after each message).

Data EK, SRK, NK, DK, NSK and N_ID are stored in non-volatile memory of the TPM, and the remaining data are stored in the sensor resource and secured using the NK key.

III. Operations in the Wireless Sensor Network with Authentication

In order to ensure proper authentication of sensors in domain and correct data transfer between sensors, in the domain should be available the following procedures:

1. Procedure for initiating M node.
2. Procedure for registering the S node in the domain of sensors.
3. Procedure for removing rM or S node from the sensors' domain.
4. Authentication procedure of the node.
5. Integration test of nodes in sensors' domain.
6. Procedure for the regeneration of S node credentials.
7. Procedure of sending data from S node to M node.
8. Procedure of reading data on M node which were received from S node.
9. Procedure for giving role rM in the domain for S node.
10. Procedure for updating resources of rM node based on resources of M node.
11. Procedure for changing the node role from rM role to role M;
12. Procedure for determining the "new" node M after the failure of the "old" node M.
13. Integration test of resources of M and rM nodes.

In this study in the following sections are comprehensively described the procedures listed in paragraphs 1-8. Other functions related to the management of the nodes being the "replicas of the master" are a subject of another study.

A. The procedure for initiating M node

This procedure is intended to create the domain of sensors and to initiate the node that will serve as the master of the domain.

Input data:

- N_ID - node identifier;
- DN - sensors' domain name;
- time periods (i.e. PR, PNR and TDV) associated with the operation of nodes rM..

The procedure for initiating M node comprises the following steps:

1. Take ownership of the TPM and SRK key generation.
2. Generate asymmetric key (NK) and symmetric key (NSK) for the node.
3. Put NK, NSK and N_ID into non-volatile memory of the TPM.
4. Generate asymmetric key (DK) for sensors' domain and put the public part of that key in non-volatile memory of the TPM.
5. Prepare of the domain description, which includes the fields DN, RN, DK, PR, PNR, TDV and then wrap this description using the public portion of the NK key. The RN field should have a content of "M".
6. Prepare of the M node description and then wrap this description using the public part of the NK key. The fields of the description should have the following values:
   N_ID = input data N_ID
   RN = „M"
   SlvK = public part of the node NK key
   NSK = the node NSK key

[3] It was assumed that Time field is modified each time the field SQ is modified. In order not to complicate the understanding of the procedures outlined in the following sections, this field has not been included in these procedures.

Stat = 0
Time = current time
SQ = random number from the range <0; 65535>.

7. Save the M node description in M node resources.

B. The procedure for registering the S node in the domain of sensors

In the procedure of registration S node in the domain is assumed that during this procedure S node is connected to the node M via the USB interface[4].

Input data:

- N_ID - node identifier;
- public part of the DK key.

The procedure for registering S node in the domain comprises the following steps:

1. Install S node in USB port of M node.
2. Take ownership of the TPM and SRK key generation.
3. Generate asymmetric key (NK) and symmetric key (NSK) for the node.
4. Put NK, NSK and N_ID into non-volatile memory of the TPM of S node.
5. Obtain the public part of the DK key from non-volatile memory of the TPM of M node and save it into non-volatile memory of the TPM of S node.
8. Prepare of the S node description and then wrap this description using the public portion of the NK key. The fields of the description should have the following values:
   N_ID = input data N_ID
   RN = „S"
   SlvK = public part of the S node NK key which is registered
   NSK = the NSK key of node which is registered
   Stat = 0
   Time = current time
   SQ = random number from the range <0; 65535>.
6. Save the S node description in M node resources.
7. Uninstall the S node from USB port of M node

C. The procedure for removing rM or S node from the sensors' domain

The procedure for removing a node is technically quite simple activity. A bigger problem is the answer to the question: under what conditions make this activity? The problems from this area that should be resolved include the following:

- Is the node is removed after one ineffective node authentication procedure (or maybe after n unsuccessful attempts)?
- After how many unsuccessful attempts to authenticate the node is removed?
- How long the node can remain in a non-active state before it is removed?

[4] If it was not possible to use the USB interface, in order to ensure the safety of the registration procedure, is required to develop additional ways of mutual authentication of both parties involved in the registration.

The following procedure does not take into account the identified problems. It is assumed that decisions on the above issues have already been taken, and the procedure can be performed only by the node M.

Input data:

- N_ID - identifier of node to remove
- Description of N_ID node recorded in the tree of trust stored on resources of M node.

The procedure for removing rM or S node from the sensors' domain comprises the following steps:

1. Prepare a remove packet (Optional[5]):

remove packet

code	id	empty	name

   where:
   code = 5 for remove packet;
   id – SQ field from description of removed node;
   empty - zeroed field;
   name -identifier of removing node (i.e. M node).
2. Wrap the remove packet with the SlvK key of N_ID node and send the packet from M node to rM or S node.
3. Remove the N_ID node description from resources of M node.

D. The authentication procedure of the node.

The procedure may be initiated by node M (authenticator) to confirm the identity of the node rM or S, and can also be initiated by the node S to confirm the identity of the node M. The procedure is based on PPP Challenge Handshake Authentication Protocol (CHAP)[18].

Input data:

- N_ID - identifier of node to check;
- Description of N_ID node recorded in the tree of trust stored on resources of M node.

The authentication procedure of the node S or rM initiated by node M comprises the following steps:

1. Prepare a challenge packet:

challenge packet

code	id	rand	name

   where:
   code = 1 for challenge packet;
   id – SQ field from description of checked node;
   rand - random number from the range <0; 65535>;
   name -identifier of checking node (i.e. M node).
2. Increment the stat field in description of checked node.
3. Wrap the challenge packet with the SlvK key of checked node (optional)[6].
4. Send the packet from M node to rM or S node.

[5] If the removed node is to be informed of the fact of the removal of that node from the sensors' domain, the steps 1 and 2 of the procedure are required.

[6] Given the limitations in terms of energy consumption and shortage of computing power of sensor you can skip steps 3, 6, 10, and 13, but then the packets will be sent in clear text and it will be the security vulnerability of the system.

5. Receive the challenge packet on checked node from M node (unwrap the packet with the private part of NK key, if needed) and prepare a response packet:

response packet

code	id	hash	name

where:
code = 2 for response packet;
id – **id** field from challenge packet;
hash - value of hash function (SHA-1) determined for concatenation of the following fields:
- **id** field from challenge packet;
- **rand** field from challenge packet;
- Symmetric Key (NSK) of checked node;

name -identifier of checked node.

6. Wrap the response packet with the private NK key of node.
7. Send the response packet to M node.
8. Receive the response packet on M node (unwrap the packet with the SlvK key of N_ID node, if needed). Verify data by comparing the value of hash field from response packet and value determined for concatenation of the following fields:
   - **id** field from challenge packet;
   - **rand** field from challenge packet;
   - NSK field from description of checked node.
9. If authentication is successful, the stat field in description of checked node is zeroed, the SQ field in description of checked node is incremented:

success packet

code	id	ok	name

where:
code = 3 for success packet;
id – **id** field from challenge packet;
ok - success message for checked node;
name -identifier of checking node (i.e. M node).

10. Wrap the success packet with the SlvK key of checked node.
11. Send the packet from M node to rM or S node.
12. If authentication fails, a failure packet is send to checked node:

failure packet

code	id	fail	name

where:
code = 4 for failure packet;
id – **id** field from challenge packet;
fail - failure message for checked node
name -identifier of checking node (i.e. M node).

13. Wrap the failure packet with the SlvK key of checked node.
14. Send the packet from M node to rM or S node.
15. In case of receiving the success package on checked node, the Sequential Number (SQ of checked node) of last send packet is incremented. In other case (i.e. receiving the failure packet or not collecting any package) the SQ of checked node is not modified.

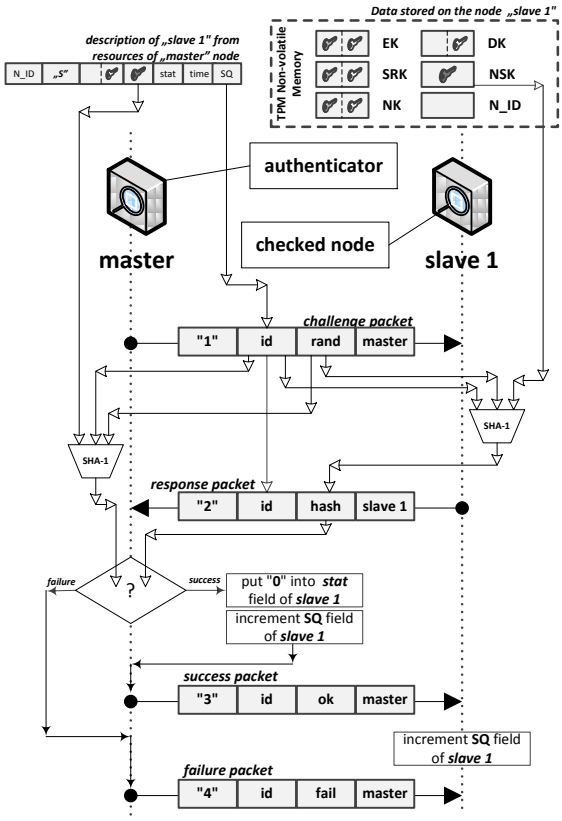

Fig. 6 The authentication procedure of *"slave 1"* node initiated by "master" node (packages are sent in clear text).

Authenticator (i.e. the node which initiates the authentication procedure) controls the frequency and timing challenges. If the above described procedure is initiated by the node S, S node takes over the authenticator role. The authentication procedure of *"slave 1"* node initiated by "master" node is showed on Fig. 6.

E. The integration test of nodes in sensors' domain

The integration test of nodes in sensors' domain is initiated by the node M. This procedure involves running the authentication procedures for all sensors' domain nodes whose descriptions are stored in the resource of node M. The procedure is to be run on demand.

F. Procedure for the regeneration of S node credentials

Procedure for the regeneration of S node credentials is initiated by node M in one of the following cases:

- overflow the sequence number SQ;
- after exceeding a fixed number of packets sent between nodes M and S;
- after a fixed time interval of validity of credentials.

The procedure can also be run on demand and then can be initiated either from the node M and node S.

Input data:

- N_ID - identifier of node to check;
- Description of N_ID node recorded in the tree of trust stored on resources of M node.

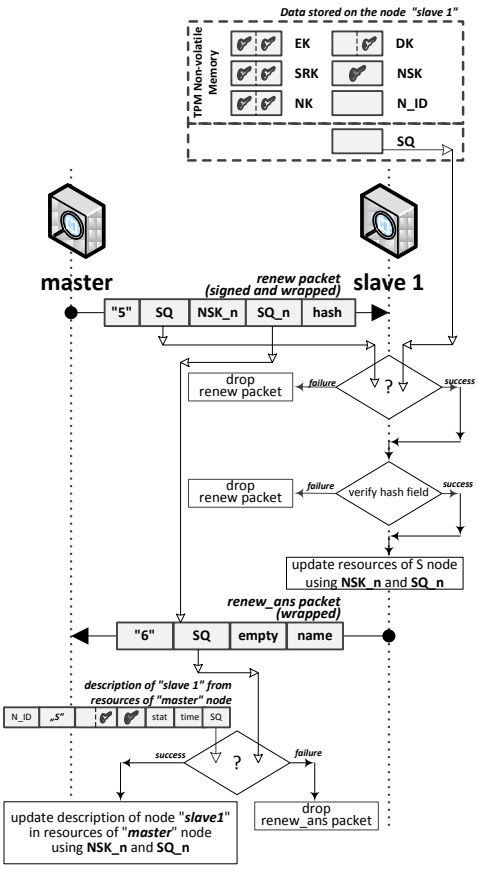

Fig. 7 The procedure for the regeneration of *"slave 1"* node credentials initiated by "master" node.

The procedure for regeneration of S node credentials initiated by node M comprises the following steps (Fig. 7):

1. Prepare a renew packet:

renew packet

code	SQ	NSK_n	SQ _n	hash

where:
code = 5 for renew packet;
SQ = current SQ incremented by 1
NSK_n – new symmetric NSK key for S node;
SQ_n – new sequential number for S node;
hash - value of hash function (SHA-1) determined for concatenation of **NSK_n** and **SQ_n** fields.

2. Sign **NSK_n**, **SQ_n**, and **hash** fields using private part of DK key.
3. Wrap renew packet using Slvk of S node.
4. Send the packet to S node.
5. Receive the renew packet on S node and unwrap the packet using private part of NK key of S node.
6. Compare SQ field from renew packet and SQ of S node. If not equal, drop packet.
7. Unsign **NSK_n**, **SQ_n**, and **hash** fields using public part of DK key.
8. Verify data by comparing the value of hash field from renew packet and value determined for concatenation of the **NSK_n** and **SQ_n**, fields.
9. If not a success, drop packet, otherwise update resources of S node using **NSK_n** and **SQ_n** fields and prepare renew_ans packet:

renew_ans packet

code	SQ	empty	name

where:
code = 6 for success packet;
SQ – **SQ_n** field from renew packet;
empty - zeroed field;
name -identifier of checking node (i.e. M node).

10. Wrap renew_ans packet using public part of NK key of S node and send the renew_ans packet to M node.
11. Receive the packet on M node, unwrap it using private part of NK key, and verify SQ field. If success, update the description of S node in resources of M node on basis of NSK_n and SQ_n fields.

G. The procedure of sending data from S node to M node

Input data:

- N_ID – identifier of node;
- SD – sensor's data
- NSK – symmetric key of S node
- DK – public part of domain key.

sensor packet

code	N_ID	SD	SQ	Hash

Fig. 8 The structure of the frame containing the sensor data

The structure of the frame containing the sensor data is showed on Fig. 8. It includes the following fields:

code = 7 for sensor packet
N_ID = input data N_ID
SD = Sensor's Data encrypted with the NSK
SQ = current SQ incremented by 1;
Hash = the value of the hash function determined on the basis of fields N_ID, SD and SQ

The procedure of sending data from S node to M node comprises the following steps:

1. Preparing of the frame containing the sensor data, as shown on Fig. 8.
2. Encrypting of the frame using the public part of DK.
3. Sending the frame to M node;
4. Incrementing SQ field in resources of S node.

H. The procedure of reading data on M node which were received from S node.

Input data:

- Received frame from S node;
- Resources of M node.

The procedure of receiving data on M node from S node comprises the following steps:

1. Receiving of the frame, as shown on Fig. 9.
2. Unwrapping of the frame using the private part of DK.
3. Searching the description of N_ID node in resources of node M. If not a success, the N_ID node is unrecognized.

4. Comparing SQ field from received frame and SQ field from node description. If not equal, the SQ is incorrect.
5. Updating the description of N_ID node:
   stat = 0
   Time = current time
   SQ = **SQ**+1
6. Decrypting of the SD field using the NSK of "slave 1" node.

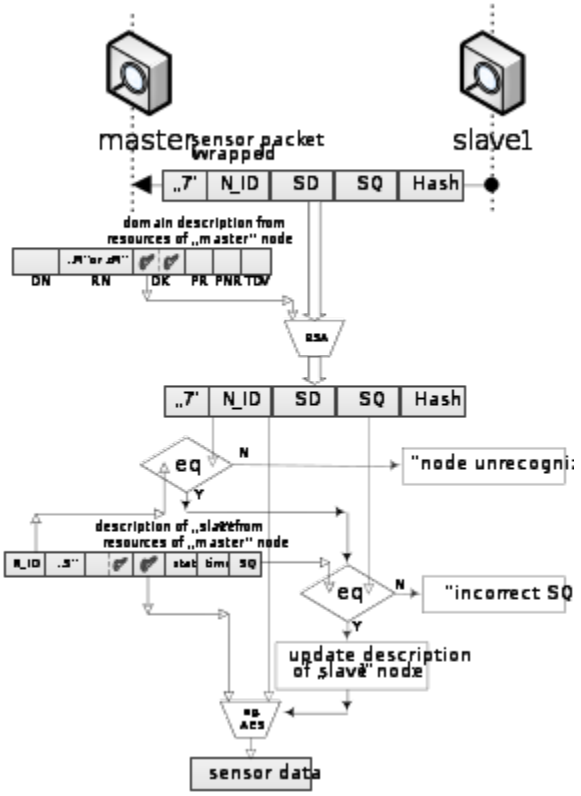

Fig. 9 The procedure of reading data on "master" node which were received from "slave 1" node

IV. Conclusion

This paper presents the model and concept of authentication in sensors' domain. For this purpose, the mechanisms provided by the TPM are used.

Solutions related to authentication elements involved in secure exchange of data require effective exchange of keys between these elements while maintaining the ability to communicate between these elements. The problem of sensors authentication is of great importance for the wireless sensor networks especially that they are mostly employed in critical areas.

On the other hand, the nature of nodes in WSNs gives rise to constraints such as limited energy, processing capability, and storage capacity. The selection of the appropriate cryptographic methods depends on the processing capability of sensors, indicating that there is no universal solution for all networks of sensors.

Taking this into consideration the TPM use is proposed for managing the root of trust and as a tool for securing the data exchange between sensors. Use of TPM will enable credentials processing by the hardware. Most of the operations is done by M node in the domain. The M node receives all the data from sensors and is used to authenticate the rest of nodes. The other nodes do not need large resources. Further work will aim at implementation of the proposed method in built model of WSN and then verifying its properties in a real environment.

References

[1] K. Sohraby, D. Minoli, T. Znati, „Wireless Sensor Networks Technology, Protocols, and Applications", Wiley, New Jersey 2007, DOI: 10.1002/047011276X.
[2] R. Faludi, "Building Wireless Sensor Networks", O'Reilly, 2011.
[3] A. Perrig et al., "SPINS: Security Protocols for Sensor Networks", Wireless Networks, vol. 8, no. 5, Sept. 2002, pp. 521–34, DOI: 10.1023/A:1016598314198.
[4] Boyle D., „Securing Wireless Sensor Networks: Security Architectures", Journal Of Networks, Vol. 3, No. 1, January 2008, pp.65-77.
[5] A. Al-Dhelaan, "Pairwise Key Establishment Scheme for Hypercube-based Wireless Sensor Networks", Recent Researches in Computer Science.
[6] Y Mohd Yussoff, H. Hashim, M. Dani Baba, "Identity-based Trusted Authentication in Wireless Sensor Network", International Journal of Computer Science Issues, Vol. 9, Issue 3, No 2, May 2012.
[7] L. Hu and D. Evans, "Secure Aggregation for Wireless Networks," Wksp. Security and Assurance in Ad Hoc Networks, 2003.
[8] B. Przydatek, D. Song, and A. Perrig, "SIA: Secure Information Aggregation in Sensor Networks," SenSys '03: Proc. 1st Int'l. Conf. Embedded Networked Sensor Systems, New York: ACM Press, 2003, pp. 255–65, DOI: 10.1145/958491.958521.
[9] W. Hu, H. Tan, P. Corke, W. Chan Shih, S. Jha, "Toward Trusted Wireless Sensor Networks", ACM Transactions on Sensor Networks, Vol. 7, No. 1, Article 5, August 2010, DOI: 10.1145/1806895.1806900.
[10] C. Krauß, F. Stumpf, C. Eckert, "Detecting Node Compromise in Hybrid Wireless Sensor Networks Using Attestation Techniques", Lecture Notes in Computer Science Volume 4572, Springer-Verlag Berlin Heidelberg 2007, pp. 203–217, DOI: 10.1007/978-3-540-73275-4_15.
[11] J. Furtak, T. Pałys, J. Chudzikiewicz, "How to use the TPM in the method of secure data exchange using Flash RAM media", Proceedings of the Federated Conference on Computer Science and Information Systems, 2013, pp. 831–838.
[12] Hu W., Corke P., Chan Shih W., Overs L., „secFleck: A Public Key Technology Platform for Wireless Sensor Networks", Wireless Sensor Networks, Lecture Notes in Computer Science Volume 5432, 2009, pp 296-311, DOI: 10.1007/978-3-642-00224-3_19.
[13] Y. Wang, G. Attebury, B. Ramamurthy, "A survey of security issues in Wireless sensor networks", IEEE Communications Surveys & Tutorials, , Volume 8, No. 2, 2ND Quarter 2006, DOI: 10.1109/COMST.2006.315852.
[14] J. Sen, "A Survey on Wireless Sensor Network Security", International Journal of Communication Networks and Information Security, Vol. 1, No. 2, August 2009.
[15] *TPM Main Part 1 Design Principles. Specification Version 1.2. Revision 116,* Trusted Computing Group, Incorporated, 2011
[16] *TCG Software Stack (TSS) Specification Version 1.2 Part1: Commands and Structures* (http://www.trustedcomputinggroup.org /files/resource_files/6479CD77-1D09-3519-AD89EAD1BC8C97F0 /TSS_1_2_Errata_A-final.pdf).
[17] S. Kinney, "Trusted platform module basics: using TPM in embedded systems", Embedded Technology Series,Elsevier Inc., 2006
[18] Simpson, W., "PPP Challenge Handshake Authentication Protocol (CHAP)", RFC 1994, August 1996.