Secure Communication in a Heterogeneous Sensor System

Ondřej Čožík, Jaroslav Kadlec, Radek Kuchta
Department of microelectronics
Faculty of Electrical Engineering and Communication, Brno University of Technology
Brno, Czech Republic
e-mail: xcozik00@stud.feec.vutbr.cz, kadlecja@feec.vutbr.cz, kuchtar@feec.vutbr.cz

***Abstract*—The article deals with security of communication in a sensor system combining a wireless RF network and Ethernet. The article deals with design of complete secure sensor system combining a wireless RF network and Ethernet. Within the suitable sensor system topology design the requirements for the fast and efficient data exchange between individual logical system layers are taken into account. One of the major requirements for our design was to develop solution with minimum computing power consumed by the communication sensor nodes. The particular system layer security is based on the known security methods and protocols (TLS protocol, improved Diffie-Hellman protocol GDH.3), which have been extended by the methods needed for their practical use (method called AVOM, which is intended for discovering and labeling of all RF network devices). A partial goal of the designed solution is to improve the robustness of the implemented security mechanism for wireless logical group security key establishment.**

Keywords-Sensor system; RF network; TLS protocol; Diffie-Hellman protocol; GDH.3 protocol.

I. Introduction

The article briefly describes the development of a sensor system including the selection of used components and also the encryption principles used in communication between devices. Wireless system security must be designed in such a way that from the beginning it is developed so as not to allow an attacker to retrieve any information from the system [1][2].

Within the research the various sensor system security methods have been observed [9][10][12][13][14][15]. Unfortunately, these methods did not meet some of our basic requirements on the developing sensor system, i.e. rate of the data exchange, computing power, synchronization, simple implementation of the secure algorithm into the microcontroller, possibility to immediately decrypt every received secured messages, and the ability to remove / add a new member to the secured sensor network.

In the survey of available methods intended to protect common data exchange in the sensor system, there has been found several security techniques, which did not meet the important demands [14][15], or meet the demands just partially[14][15]. An overview and comparison between different security methods including their features can be found in [18]. Since this security systems are not appropriate, the logical step was to create a security system focused on the implementation simplicity, the rapid encryption key determination for all devices in the sensor network, device access to the system management capability and minimum traffic load necessary for additional information transmitting. The proposed system will have two modes – the initialization mode and the normal mode. During the initialization mode, the encryption key will be provided by below mentioned mechanisms. Subsequently, when the normal mode become active, the messages are encrypted using the agreed key and can be sent into the sensor system.

The encryption key distribution within the higher hierarchical system layer is based on the TLS protocol [2], extended Diffie-Hellman protocol GDH.3[3] which is the crucial part of the encryption key distribution in the wireless RF network.

From the outset, demands on the topology of the sensor system corresponding to their planned use are mentioned. Also, the main requirements for particular devices in the system are specified. For both the selected topology and each component the pros and cons are outlined.

The requirements for simplicity and speed of encryption/decryption are especially important, because every algorithm will be implemented into the microcontroller. In this part of the article, the provision of a secret key and the onward transmission of the encryption key from the top system layer down to the lowest layer is illustrated. Following this is the assessment of the designed solution in terms of communication security and the time demands for the encryption / decryption process.

The final section of the article deals with the selection of a safe and relatively simple encryption method for communication between devices in a wireless RF network and also at the Ethernet level.

II. Required System Topology

The proposed topology of the complete system is adapted for rapid message exchange between master and slave devices. During the exchange, relatively large data flows between certain devices may be included (messages size is up to tens of bytes in RF network). Furthermore, emphasis was placed on the possibility of an accurate synchronization between the wireless modules and control device.

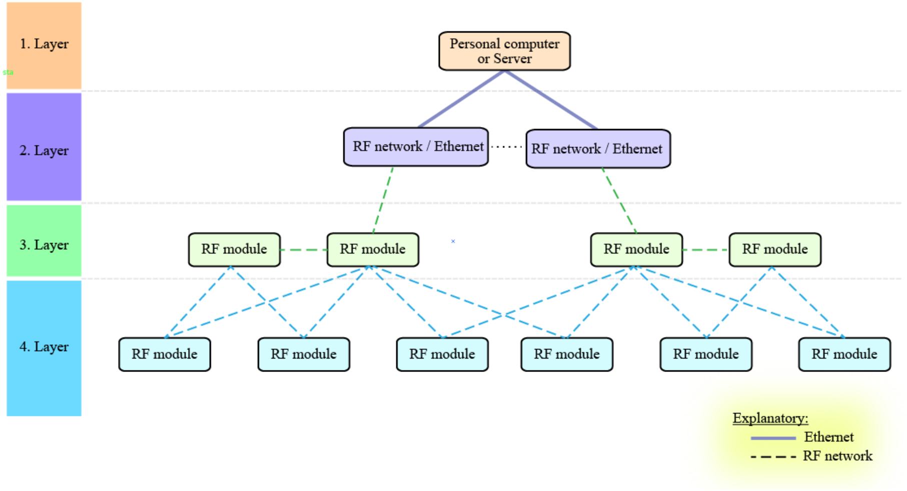

Figure 1. Proposed sensor system hierarchy

The resulting topology is presented in Figure 1 and it can be seen, that the system topology is divided into the four layers. The control device, which could be, e.g., server or personal computer with appropriate control software installed, is in the top (first) system layer and it processes packets over Ethernet. The second system layer of created topology contains a converter placed between the Ethernet and wireless RF network. In this layer, there could be more than one converter, but only one converter for one logical group of the RF modules. The main task for them is to forward data messages from the wireless network to the Ethernet and vice versa. The converters should be positioned correctly to avoid overloading any RF module at the lower level. A suitable compromise between the number of transmitters (converters) and wireless modules in the group, which can use it, must be found (it depends on node message size and message exchange frequency between the nodes and server).

The third layer represents the RF master modules, which are fixed in the area, serving as a messages repeater to a desired device and back.

Finally, in the lowest (fourth) layer there are portable wireless RF slave modules periodically sending data and status information to the parent device in the hierarchy (sensors). Therefore, it is desired to establish secure communication in order to avoid a leakage of sensitive information transmitted in the system or misuse of invalid data by a possible attacker that could cause an error in the sensor system, or even cause it to malfunction.

Portable RF (slave) modules transmit data to the third layer, that takes care of transferring messages to the RF module, which is able to directly communicate with the converter between the RF network / Ethernet in the third layer.

The messaging system for one RF slave module in the fourth layer is shown in Figure 2. One of the requirements for the proper function of the system is RF modules in the fourth layer should always be available for at least two RF master modules.

Since the RF slave modules are portable, there is a complication with security options due to the possibility that a module may be in the area of the first RF master module at one moment, but in the next moment it could be in another RF master module area. Because of this feature, the system must be secured in a manner allowing all devices in the third layer to decrypt the message from every device in the fourth system layer. A similar situation occurs between the second and the third layer in the hierarchy of the sensor system, the only difference is, that the modules in those layers will be moved very rarely.

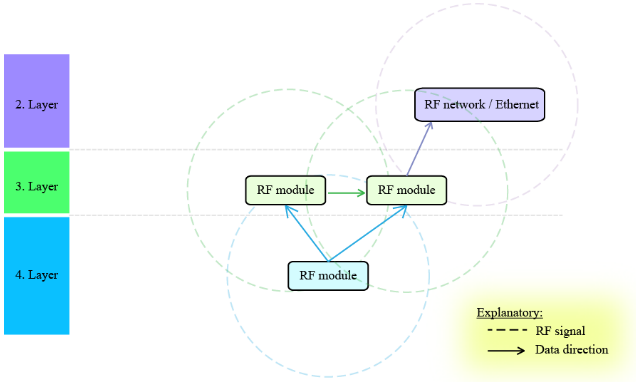

Figure 2. Message forwarding at lower sensor system layers in the RF network

A method of message forwarding between devices in the third layer is shown in Figure 2. Due to sensor system price reduction it is necessary to have the lowest number of converters in a sensor network. To make a successful transmission from a desired RF master module to the converter device, the message has to be sent from the sender towards the converter. If the converter is not in the sender's area, the message will be forwarded towards it (in the third layer) until the converter receives the sender's message.

III. Secure Communication on an Ethernet network

Transport layer security (TLS) protocol for the security of communication in the highest level in the sensor system hierarchy will be used [3][4]. TLS protocol is a free version of SSL protocol. Using the mentioned protocol, it is possible to create encrypted communication channel between the converters and the server via Ethernet.

A. Communication establishment using the TLS protocol

A communication establishment has to be performed before the two parties are able to transfer data via a secured channel using the TLS protocol. An identity and other information has to be exchanged between server and client to create the encrypted channel and then secure communication can begin.

The connection establishment (handshake) includes a total of four consecutive phases [1]. The description of each handshake stage is not included in the article, because the TLS protocol in general is well known and used.

Once the handshake process between the client (converter from RF network to the Ethernet) and the server has been completed, the data exchange between two devices, that are authenticated and have the necessary keys, can be initialized.

B. A Data exchange via TLS protocol

After a successful TLS handshake protocol between the transmitter and server, both sides are able to encrypt communication using the agreed key. The sending procedure for application data via secure communication channel is shown in Figure 3.

In the first step, the user’s secret data are taken from the application layer and divided into blocks with a maximum size of 2^{14} B (according to the TLS protocol specification). In the second step, a lossless compression function can be applied to the separate data blocks from the previous step. The compress function between both sides was arranged in the handshake protocol, in the current TLS protocol version there is no compression method by default.

In the next step, a message authentication code is added to the encrypted and compressed data block. The code is determined by the HMAC technique [1].

In the fourth step, the compressed segments with the authentication code are encrypted using the selected algorithm. The AES encryption algorithm [6] will be used in the proposed system due to availability of an AES hardware encryption module in the Texas Instrument microcontroller named CC430F5137, which is contained in all designed RF devices. The encryption key length in these devices could be 128 b (it is sufficient length for this kind of sensor network).

In the last step, a 5 B header to the encrypted data block is added. The first byte of the header represents the protocol version used for attached data processing. The next two bytes contain the major and the minor version of the used protocol and the last 2 bytes carry the entire encryption segment length.

The communication between all other devices on the second layer of sensor system topology and server is established in the same way. When all devices on the second layer are securely connected to the server, it is necessary to create an encrypted connection also on the lower system layers. This issue is the subject of the next section IV: Secure communication in the wireless section of the sensor system.

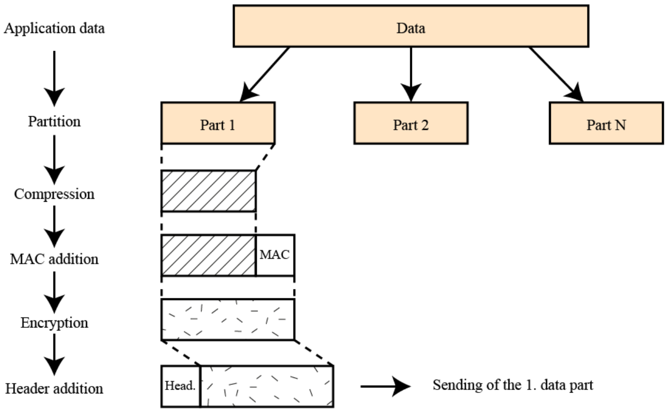

Figure 3. Application data encryption in the TLS protocol [1]

IV. Secure communication in the wireless section of the sensor system

In this section, the security of communication between the first and second layer is not considered, this has been described in the section 3. Only secure communication from the second system layer below is taken into account, i.e., RF network security.

For the actual communication design between the devices it is important to analyze all potential attacks and feasible security risks for this type of system and the most suitable security method should be chosen [7][8][9].

A. Potential risk and related problems

In wireless networks generally, there are many ways of how to attack system security [1][3]. One of them may be listening to network communication. In the case of an unsecure network, an attacker can read all transmitted messages. The solution, which removes this problem, may be encryption of all messages in the designed RF network. Another difficulty could then emerge – how to manage the encryption keys for all wireless devices?

Following this we must assume that the wireless RF system is already secured. Although the attacker can intercept the transmitted cipher, without the used encryption key the cypher is then irrelevant. Data collected in the sensor system are from a large number of RF measuring modules. They send this data to the parent layer in the network hierarchy. Occasionally the measured value changes very slowly (or not at all) and the module will send the same value over and over. An attacker can take advantage of this information and break the encryption. To prevent this sensor system feature, the *COUNTER* field will be added to all messages. When the RF module sends the measured value, it immediately increments the *COUNTER* field (1).

$$COUNTER_{(t+1)} = COUNTER_{(t)} + 1 . \qquad (1)$$

The advantage of a block cypher is that a change of a single bit at the encryptor input causes a large change at its output. Basically, it is possible to achieve a completely different cypher even with two identical data frames sent via an RF module at the lowest level, only with a varying $COUNTER$ field.

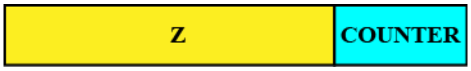

Figure 4. Message Z in RF network with a COUNTER field

The situation of the adding a $COUNTER$ field to the message is shown in Figure 4. The field $COUNTER$ will be added to the permanent data field Z . This precaution should also reduce the possibility of re-sending a previously intercepted message from the attacker. This is because the transmitter and receiver know the current $COUNTER$ parameter value and receive the message only if the parameter from the message is identical to its $COUNTER$ value. If any attacker captures a packet and subsequently sends it, the receiving party will assess the message as invalid, because the same parameter has already been received and the $COUNTER$ value is different.

To prevent other types of attacks such as a brute force attack, or an attempt to capture as many messages as possible in order to determine the encryption key, the proposed security feature allows the encryption key change in a secure way. All microcontrollers used in all devices on all layers, except the highest one, contain AES module with the option of a 128 b key [4][5]. It is appropriate to use this module for standard encryption. Presently, the main problem is how to securely set up the key in all devices.

There is a simple way to solve this problem, if the communication is only between two sides, as shown in Figure 5.

Figure 5. Two parts communication (Device A and B)

In this case, it would be sufficient to use the Diffie-Hellman (DH) key establishment protocol [6]. The protocol deals with an ideal two sided communication. This situation does not appear in the proposed sensor system and it is necessary to securely establish the encryption key for multiple devices [7][8][9]. One of the suitable protocols, which can be used in the sensor system, is named GDH.3 [10].

B. *The algorithm for automatic detection and labeling of modules in an RF network*

In order to use the GDH.3 protocol in the system, it is necessary to specify a logical group of modules that have the same secret key used for device identification. When all wireless RF devices prove their identity, the parent device will determine a new encryption key for the whole group. The encryption key will be sent using the previous established secret group key. For the execution of the GDH.3 protocol, each module in the logical group must have a unique number (address).

The unique number inside the group must be assigned automatically due to adding a new wireless module and avoiding collision with another previously labeled device. For this purpose a method of automatic detection and labeling of the wireless device (AVOM) was designed.

For correct search functionality it is necessary to send a token in the RF network. The device assigns the unique numbers (addresses) to the new identified modules in its communication area. Along with the token, the highest assigned address will also be transmitted. Due to this mechanism, the next device knows exactly the following address, which can be assigned to the new devices and there will be no address collision in the RF network.

The resulting AVOM method, which was used in the system hierarchy, is shown in Figure 6. It should be noted, that before the start of this function, all devices have to know about the new sequence of searching and labeling devices. This notification in the RF network can be done using a broadcast message, i.e., message delivered to all devices in the network. On the basis of that message, all devices delete their current addresses, related information and stop all further communication until the system is completely secured (it can be sent using a broadcast message again).

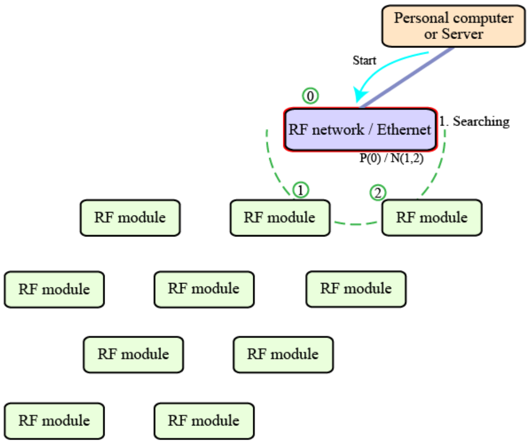

Figure 6. Basic system hierarchy for an address determining - includes a first scanning step in the RF network

In Figure 6, we can see the typical hierarchy of the proposed sensor system. In this illustration is also shown the first step of the searching method. Firstly, the server sends the AVOM start command to the RF / Ethernet Converter in the second layer, which is on the highest layer from all the RF devices. Therefore, the RF / Ethernet converter selects the address 0 (in Figure 6 is the assigned address above the module in a green circle). Secondly, the scanning of all available devices in the converter area is launched. When the converter gets all directly available devices via RF communication, the converter individually assigns to these devices their addresses according to a chosen criteria (e.g.

signal strength, response time, etc.). In Figure 6 addresses 1 and 2 are assigned.

Each device will always save the parent device address (address of the device sending the token) and all unlabeled devices in its area to which the token has not been forwarded yet. In the figures these numbers are always written below the module. For example, the RF / Ethernet converter in Figure 6 has the previous address device equals $P(0)$, this means there is not a device above it in the system hierarchy. The algorithm is able to recognize, that all devices have been labeled. Numbers $N(1,2)$ mean, that the token has not been sent to the device with address 1 and 2 yet.

The module highlighted in red (in the figures) currently has the token and is allowed to label the modules in its RF communication range; it is indicated by the green dashed line.

Thirdly, the token is passed to the next device in the network (with address 1). The token is always sent to the device with the lowest newly assigned address. When a selected device receives the token, it saves the address of the previous device (0), then scans its communication area for new unlabeled devices and assigns them appropriate addresses, i.e., 3, 4 and 5 in Figure 7. The module holding the token saves in its memory all newly labeled devices to send them the token in the future. The device with the lowest address (3) is chosen and the token is forwarded to this device.

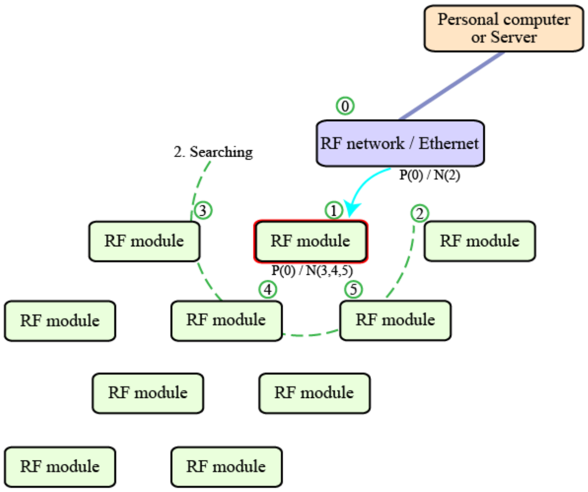

Figure 7. Second step of the AVOM function

The same procedure is applied until the algorithm arrives at the device in its communication area where there is no an unlabeled module. This scenario is shown in Figure 8, the token holds the device with address 7 and in its range there are only labeled devices with addresses 8, 9 and 10, which are final. The device with address 7 subsequently forwards the token to the final devices 8, 9 and 10. Each of the final devices checks its communication area, but there is no new device, so the token is sent back to the device with address 7 and so on. The forwarding process is shown in the figures by arrow. The numbers near the arrows represent the individual steps of token forwarding. When the device 7 verifies all 3 devices in its area, there is no device to forward the token, so it sends the token back to the parent device with address 6. The device with address 6 also does not have any device in its area and sends the token back until the token arrives back at the device with address 1.

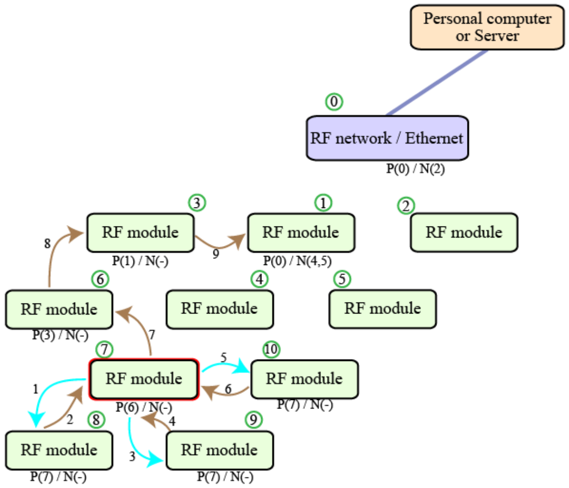

Figure 8. The end of the forward phase of the AVOM algorithm

The device 1 has 2 modules stored, which have not received the token, so it sends it immediately to the module with a lower address, i.e., 4. When a new device is not found within its communication range the token is sent back to the device with address 1. It stores the last device, which still has not received the token. The same procedure is undertaken with the device with address 5. This module also has an empty queue and returns the token to the device with the address 0 (RF / Ethernet converter). This situation is shown in Figure 9.

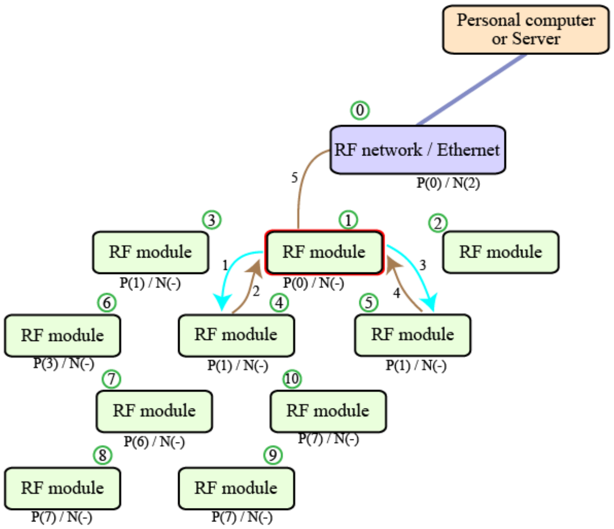

Figure 9. Token forwarding during the AVOM function

The module with address 0 (converter) still has one device in its queue, which has not received the token yet, specifically it is the RF module with address 2. When the token exchange is complete, the new address is assigned to all identified devices in the RF network. Since the address of the previous module is identical to the converter, the converter sends a message to the server with a device numbering completion announcement in the RF network (see in Figure 10).

When the labeling process required for the GDH.3 protocol startup is completed, a temporary secret key will be established in the labeled logical group for precise device identification and subsequently encryption key transmission.

C. A group key arrangement using the GDH.3 protocol

In the basic Diffie – Hellman protocol the key is placed between the member M_1 and M_2 using several steps. In the first step, member M_1 sends the value u according to (2) to member M_2 [1][6].

$$u = \alpha^{N_1} \left(\mathrm{mod}\ p\right). \tag{2}$$

In (2), there is a group generator α, a random number N_1 is generated by M_1 and p is a prime number. The values p and α are known to both sides.

Once the member M_1 sends the value u, the member M_2 sends another value v to the member M_1. A computing v value describes (3).

$$v = \alpha^{N_2} \left(\mathrm{mod}\ p\right). \tag{3}$$

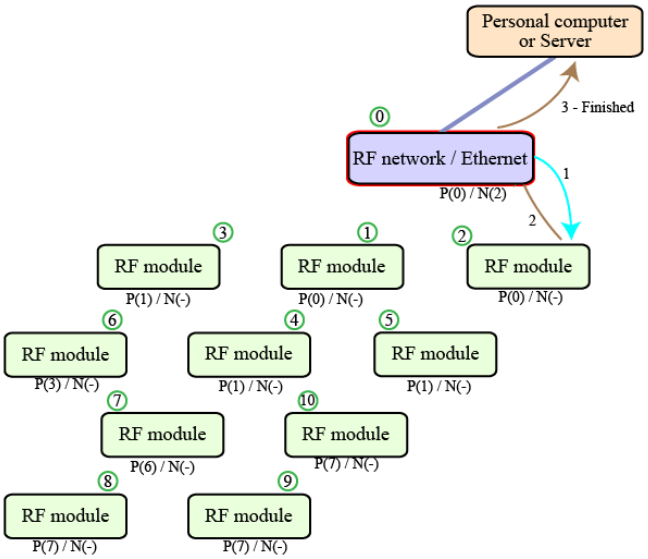

Figure 10. Final phase of the AVOM function

Equation (3) is similar to (2), except there is a different coefficient denoted as N_2. When both sides exchange their values, they are able to determine the same secret key K. The first member M_1 will use (4) and the second member will use (5). Now, both of them have the same secret key K.

$$K = v^{N_1} = \left(\alpha^{N_2}\right)^{N_1} \left(\mathrm{mod}\ p\right). \tag{4}$$

$$K = u^{N_2} = \left(\alpha^{N_1}\right)^{N_2} \left(\mathrm{mod}\ p\right). \tag{5}$$

To establish a group key the extended Diffie – Hellman protocol (generally for n devices) has to be used [3].

During the initialization process of the entire protocol, which allows determination of the shared key it is necessary to assign the address to all devices in the group. The devices without the address will not be able to determine the new key (for more information, see section IV.B The algorithm for automatic detection and labeling of modules in an RF network). As in the basic DH protocol, all devices know the public parameters denoted p and α. In addition, each group device M_i must generate its own random exponent N_i. Because of simplicity, the operation $\mathrm{mod}\ p$ will not be shown in the next equations.

In the first stage of the protocol, the first module M_0 will generate a value α^{N_0} using its secret exponent N_0 and sends the value to the module with the following address (device M_1). The device M_1 computes the value $\alpha^{(N_0 \cdot N_1)}$ and transmits the new value to the next device M_2. The procedure is repeated until the transmitted value reaches the device M_{n-2}, where n is total number of modules in the group. The value is then also transferred to the device M_{n-1}, which calculates the last value, but does not send it to the last device M_n. Generally, it is possible to determine the computed value u_k in the device M_k by (6) [3].

$$u_k = \alpha^{\prod_{k=0}^{i} N_k}. \tag{6}$$

In the following (second) stage of the protocol GDH.3, the broadcast message with the value computed by the module M_{n-1} is sent. All modules, including the module M_n, receive the message containing a value u_{n-1} specified by (7) [3]. The last module M_n has to save this value due to potential extension of the group.

$$u_{n-1} = \alpha^{\prod_{k=0}^{n-1} N_k}. \tag{7}$$

During the third stage, each device M_i receives the broadcast message with the value (7), and subsequently excludes its own random exponent N_i by extraction of the root (7) by inverse value of the exponent N_i^{-1} and the result

ISBN: 978-1-61208-399-5

is sent to module M_n. Generally, the sent value u_i from the module M_i can be described by (8) [3].

$$v_i = \alpha^{\prod_{k=0}^{n-1} N_k} \mid k \neq i \,. \tag{8}$$

In the fourth stage, module M_n must save all received values, raise them by its random exponent N_n and send them using the broadcast message again. An individual message contains the value s_i done by (9) [3].

$$s_i = \alpha^{\prod_{k=0}^{n} N_k} \mid k \neq i \wedge i \in [1, n-1]. \tag{9}$$

Each module M_i obtains the value s_i in this stage. When the module M_i uses again the random exponent N_i on the received value s_i, it computes the secret group key K, which will be used for exact module identification and subsequently for distribution of the communication encryption key.

$$K = \alpha^{\prod_{k=0}^{n} N_k} \,. \tag{10}$$

The value of secret group key K , which was established using the GDH.3 protocol, can be determined by (10) [3].

D. Adding a member to the group with a secret key

Over the sensor system`s lifetime, there could be a requirement for a system expansion by adding a new module into the existing group with the secret key. All the devices communicate using the established encryption key, but the values for computing the secret group key are stored within it. The new group key, which will also be used for the new member, is based on the stored values.

The group member M_n with the last address must store the values from the second and the third stage of the group key establishment. Initially, the last member M_n will generate a new random exponent $\overline{N_n}$, which raises the second stage stored value by the new exponent $\overline{N_n}$ and obtains a value defined by (11) [3].

$$\alpha^{\prod_{k=0}^{n} N_k} = \alpha^{N_0 * \ldots * N_{n-1} * \overline{N_n}} \,. \tag{11}$$

The module M_n sends the value (11) to the new device M_{n+1}, which generates its own exponent N_{n+1} and computes a new secrete key K_{n+1} for the whole group (12) [3].

$$K_{n+1} = \alpha^{\prod_{k=0}^{n+1} N_k} = \alpha^{N_0 * \ldots * \overline{N_n} * N_{n+1}} \,. \tag{12}$$

The final stage of adding a member is that the device M_{n+1} calculates n new values obtained from the device M_n. Into these values it has to add a generated exponent N_{n+1} and send them out by broadcast messages to allows other modules to determine the new group key K_{n+1}. Basically, the third and the fourth GDH.3 protocol stage is executed once more.

$$\alpha^{\prod_{k=0}^{n} N_k} \mid k \neq j \wedge j \in [1, n]. \tag{13}$$

The module M_{n+1} sends the value defined by (13) to the rest of the devices. One exponent N_j is missing in each of the sent values so it can be completed only by the device M_j.

E. Removing a group member

Due to security reasons, the algorithm has to have the ability to remove a particular device from the group. The device M_n is important for removing the device M_p (where $p \in [1, n-1]$), because all values are saved in it from the fourth stage of group key K establishment. The module M_n must generate a new random exponent $\overline{N_n}$, which will be used for the calculation of the $n-2$ values according to (9).

$$K = \alpha^{N_0 * \ldots * N_{p-1} * N_{p+1} * \ldots * N_{n-1} * \overline{N_n}} \,. \tag{14}$$

A new value for the device M_p is omitted so it is not possible to determine the new secret key K for this module in the future (14) [3].

The removal of M_n, device M_{n-1} takes over the role of the last module with the designation M_n. It also stores all the data from the fourth phase of the key establishment. Firstly, exponent N_{n-1} is cleared of stored messages (12) and it generates a new exponent N_n, and uses it to calculate new values (12), which sends the results to all devices in the group. Since the random coefficient N_n and N_i (according to recipient M_i) are missing in all messages, therefore, the last module M_n is not able to determine the resulting communication key K.

F. Portable RF modules security at the lowest system layer

The lowest layer (fourth) in the hierarchy of the sensor system contains only the portable RF modules communicating with RF devices at the higher layers. From the requirements on the sensor system it follows, that each portable RF module has to be reachable from more than one higher layer RF device. It is obvious, that communication with more than one encryption key would be too difficult due to key management.

The final encryption key will be transferred to the portable device using DH protocol between the module and the server after a successful authentication of both parties. The authentication will take place immediately when the device sends the access request to the network. The portable device creates a stipulated request and encrypts it using devices private key SK_m. The created cypher M_s device encrypts once more with the server's public key VK_s, which is available to all RF modules.

The outcome is that the encrypted message M is created and is sent through the transmission channel to the server. The entire encryption process of the requirement is captured in Figure 11.

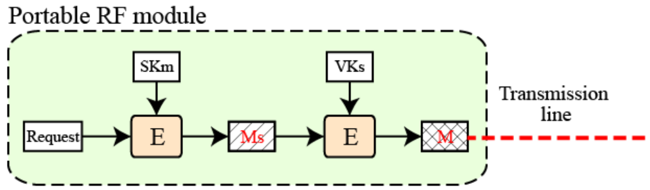

Figure 11. Request encryption in the portable RF module for sensor network access

The server receives a message M and applies its private key SK_s and gets the message M_s. The server stores all the ID's of all portable RF modules, which are allowed to access the RF network. These records can be edited by an authorized person with access to the server databases only. In the database, along with the ID the public keys VK_m are saved.

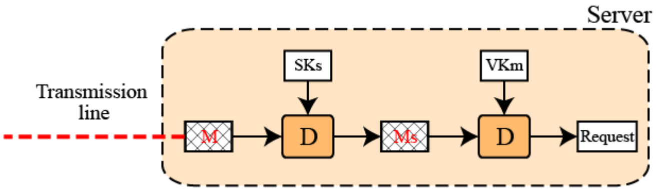

Figure 12. Message request for sensor network access and decryption in the server

The server is able to apply the proper public key associated with the module requiring access to the network. Thus, the server can decrypt the original message (see Figure 12).

When the server receives a valid request, it generates a random value for the key establishment according to the DH protocol between two devices (2), and encrypts the result in reverse order (firstly it uses its private key SK_s and following this encryption by the portable RF module public key VK_m). The message with an encrypted random value is forwarded back via the transmission channel to the RF module.

The message in the RF module is sequentially decrypted using the private key SK_m and the public key VK_s. The device generates another random number. It describes (3). The number is appropriately encrypted (as in the first request) and sent to the server (see Figure 11). In this way, the possibility of an attack by the man in the middle is excluded. This type of attack could occur only by sending unencrypted values for the DH protocol.

Now, both participants are able to determine a shared key, which is used only for transferring the final encryption key. All devices in the RF network have the final encryption key and using this key, they are able to communicate with all the RF devices at the third sensor system layer. When the final key is transferred to the RF module, it is allowed to start full communication with all devices with an RF interface across the RF network and it is guaranteed, that in the case of a network security breach, there is a possibility of how to securely establish a new encryption key without changing the firmware of each device.

V. Conclusion

In the article, the proposed sensor system is described including the necessary requirements for proper functioning of the system. The system topology of the sensor system and the communication principle at various levels of the system was described.

In the second part of the article, the secure communication possibilities at the highest level in the hierarchy of the sensor system for the Ethernet network were discussed. For Ethernet security, the TLS protocol was chosen. The basic principle of secure communication establishment and message encryption in the TLS protocol was also referred to.

The third part of the article deals with the security of the wireless section of the sensor system. Firstly, the wireless network scanning and address assignment to the individual RF modules in the second and the third layer was demonstrated in detail for the group key negotiation. For the group key arrangement, the GDH.3 protocol was used. The protocol allows adding another member to the group that was already established, as well as the removal of any group member. Through the negotiated group key, the server forwards to all RF modules the encryption key, which will be used for normal communication encryption (data in the RF network will be encrypted using the AES algorithm with a 128 bit key length).

At the lowest (fourth) model hierarchy layer, which contains the portable RF devices, initially, it was necessary to choose an authentication method for these devices and subsequently, upon successful authentication, the connection is established with the requesting device (DH protocol). After the establishment of a secure connection, the final encryption key is sent to the RF module.

The method of determining the encryption key in the proposed sensor system was designed and also illustrated.

Acknowledgements

Research described in this paper was financed by the TA04010476 project TACR "Secure Systems for Electronic Services User Verification" and by Czech Ministry of Education in frame of National Sustainability Program under grant LO1401. For research, infrastructure of the SIX Center was used.

REFERENCES

[1] W. Stallings, “Cryptography and Network Security: Principle and practice”, 5, Boston : Prentice Hall, 2011. 719 pages, ISBN 01-360-9704-9.

[2] T. Dierks and E. Rescorla, “The Transport Layer Security (TLS) Protocol”, Version 1.2. [Online] August 2008, [Cited: 28. 2 2015.] http://tools.ietf.org/pdf/rfc5246.pdf.

[3] M. Steiner, G. Tsudik and M. Waidner, “Diffie-Hellman Key Distribution Extended to Group Communication”, [Online] 1996, [Cited: 13. 2 2015.] http://corsi.dei.polimi.it/distsys/2007-2008/pub/p31-steiner.pdf.

[4] J. R. Vacca, “Network and System Security”, Burlington : Syngress/Elsevier, 2010, 368 pages, ISBN 15-974-9535-2.

[5] NIST Computer Security Division (CSD), :FIPS 197, Advanced Encryption Standard (AES)”, [Online] 26. November 2001, [Cited: 3. 2 2015.] http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

[6] Texas Instruments, Inc. “AES128 - A C Implementation for Encryption and Decryption”, [Online] A, March 2009, [Cited: 25. 2 2015.] http://www.ti.com/lit/an/slaa397a/slaa397a.pdf.

[7] J. F. Raymond and A. Stiglic, “Security Issues in the Diffie-Hellman Key Agreement Protocol”, [Online] 2003, [Cited: 16. 2 2015.] http://crypto.cs.mcgill.ca/~stiglic/Papers/dhfull.pdf.

[8] Y. Kim, A. Perrig and G. Tsudik, “Tree-based Group Key Agreement”, [Online] 2002, [Cited: 18. 2 2015.] http://www.ics.uci.edu/~gts/paps/kpt04a.pdf.

[9] K. Stewart, T. Haniotakis and S. Tragoudas, “A Security protocol for sensor networks”, Illinois : GLOBECOM '05, IEEE , vol.3, 2005, pp.1827-1831.

[10] G. A. Jolly, “Low-Energy Key Management Protocol for Wireless Sensor Networks”, [Online] 2002, [Cited: 3. 2 2015.] http://www.gta.ufrj.br/wsns/Security/LowEnergykey.pdf.

[11] H. Chan, A. Perrig and D. Song, “Random Key Predistribution Schemes for Sensor Networks”, [Online] 2003. [Cited: 2. 2 2015.] http://repository.cmu.edu/cgi/viewcontent.cgi?article=1025&context=ece.

[12] E. Shi and A. Perrig, “Designing Secure Sensor Networks”, December 2004, IEEE Wireless Communications, 2004, pp.38-43, ISSN 1536-1284.

[13] S. J. Jang, “A Study on Group Key Agreement in Sensor Network Environments Using Two-Dimensional Arrays”, [Cited: 28. 2 2015.] http://www.mdpi.com/1424-8220/11/9/8227, 2011, pp.8227-8240, ISSN 1424-8220.

[14] A. Perrig, “SPINS: Security Protocols for Sensor Networks”, [Online] 2002, [Cited: 21. 2 2015.] http://www.csee.umbc.edu/courses/graduate/CMSC691A/Spring04/papers/spins-wine-journal.pdf.

[15] A. Perrig, “The TESLA Broadcast Authentication Protocol”, [Online] 2002, [Cited: 21. 2 2015.] http://users.ece.cmu.edu/~adrian/projects/tesla-cryptobytes/tesla-cryptobytes.pdf.

[16] Texas Instruments, Inc. “C Implementation of Cryptographic Algorithms”, [Online] August 2012, [Cited: 28. 2 2015.] http://www.ti.com/lit/an/slaa547/slaa547.pdf.

[17] E. Bresson, D. Pointcheval and O. Chevassut, “Group Diffie-Hellman Key Exchange Secure against Dictionary Attacks”, [Online] 2002, [Cited: 2. 2 2015.] http://www.di.ens.fr/~bresson/papers/BreChePoi02c_full.pdf.

[18] Y. Wang; G. Attebury, B. Ramamurthy, "A survey of security issues in wireless sensor networks," Communications Surveys & Tutorials, IEEE , vol.8, no.2, 2006, pp.2,23, doi: 10.1109/COMST.2006.315852.