Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 536718, 9 pages
doi:10.1155/2012/536718

Research Article

One-Time Broadcast Encryption Schemes in Distributed Sensor Networks

Pawel Szalachowski[1] and Zbigniew Kotulski[1,2]

[1] *Institute of Telecommunications, The Faculty of Electronics and Information Technology, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland*
[2] *Institute of Fundamental Technological Research of the Polish Academy Siences, Pawinskiego 5B, 02-106 Warsaw, Poland*

Correspondence should be addressed to Pawel Szalachowski, p.szalachowski@stud.elka.pw.edu.pl

Received 12 July 2011; Revised 8 December 2011; Accepted 10 December 2011

Academic Editor: Yuhang Yang

Broadcasting is a message-transferring method characteristic for majority of sensor networks. Broadcast encryption (BE) is broadcasting encrypted messages in such a way that only legitimate nodes of a network can decrypt them. It has many potential applications in distributed wireless sensor networks (WSNs) but perfect deploying of that method is very difficult. This is because of a WSN is a very dynamic network which includes nodes with limited computational, storage, and communication capabilities. Furthermore, an attacker in this environment is powerful. He can eavesdrop, modify, and inject messages or even capture a large number of nodes, so the solutions must be both secure and efficient. This paper describes several BE schemes from the point of view of WSNs. We present in details the schemes called *onetime*, and we show how these methods can be applied in distributed sensor networks. We mainly focus on data origin authentication and rekeying processes, crucial for security in such a hostile environment. An analysis and evaluations of proposed schemes are also provided.

1. Introduction

The popularity of WSNs is an effect of recent advances in communication and development of the microelectromechanical systems (MEMS) technology. Now we have low-cost, low-power, and small devices which are mainly used for sensing, measuring, gathering, and then transmitting data from distributed locations (or an environment) to some acquisition center. Potential applications of WSNs are military (surveillance, command control, reconnaissance, intelligence, etc.), environmental (flood or fire detection, pollution transport, hostile objects observation, etc.), medical (elder people or patients monitoring), and other home and commercial applications. Surveys on the sensor network technology and its applications are presented in, for example, [1–3]. WSNs need communication protocols with special properties. Such approaches must be self-organizing [4], self-healing, fault tolerant, and secure. It is difficult to achieve all these features for devices with as limited hardware resources as wireless sensors are. Therefore the protocols must be additionally lightweight.

Broadcast encryption is an ideal proposition for WSNs. Sending encrypted messages which can be decrypted only by a predefined group of nodes can be very helpful in a WSN. Additionally, with this method we can manage strictly selected nodes [5] or configure them. Furthermore, by BE, we can fix a group of nodes and equip them with a shared key limited to just this group. Since then participants of the group can communicate each other in a secure way. However, BE must be realized in a very efficient and secure way and this is the crucial problem of BE. There are BE schemes using symmetric and asymmetric (public key) cryptography; in this paper, we focus mainly on the symmetric cryptography approaches. As we already mentioned, a typical node in a WSN has very limited resources. A small battery and weak computation efficiency make usage of the public key cryptography (PKC) impracticable. Although there are many BE schemes based on PKC, in further considerations we will not focus on them in this paper. A lightweight PKC is an objective of many research papers, but its implementations are too slow for many applications, see for example, [6–8]. Furthermore, an imprudent application of PKC (or of

some other expensive method) can make a wireless network susceptible to denial of service (DoS) attacks. Another crucial resource is memory. Constructing a security protocol, we must minimize amount of a key-storage memory. Next hardware constraint is the radio bandwidth. Because of low-transmission power, messages sent and received should be as short as possible, which gives an advantage to the symmetric cryptography.

In contrast, adversaries in these environments are very powerful [9, 10]. Eavesdropping, modifying, replying, and injecting packets are very easy when radio waves are used as a medium. Moreover, we should assume that an adversary can physically capture a number of sensors or can take control over them. Hence, the BE schemes for WSNs should be resistant to these types of attacks or at least should discover them.

The rest of the paper is organized as follows. We define the BE (with related issues) and present some significant approaches in Section 2. In Section 3, we give detailed overview of the schemes called *one-time schemes*. Some improvements of these schemes we present in Section 4 and next we analyze them in Section 5. Finally, in Section 6 we give conclusions and propose future research.

2. Broadcast Encryption

BE is some special class of key distribution schemes, which belongs to the conference key distribution protocols. Its goal is to allow a broadcast center (BC) to distribute an encrypted content to an arbitrary dynamically changing set of destination nodes. Only these nodes are able to decrypt messages. Let U be a set of all nodes in a WSN, T be a set of privileged (destination) nodes, and $S = U - T$ be a set of unprivileged nodes. Each member of U is equipped with a secret key (or, in some protocols, with several secret keys) shared with the BC. This key is called the preshared key (PSK). Further in this paper, we use the terms "node" and "user" interchangeably. They both denote a regular sensor in the network.

The BC in a BE scheme uses two types of messages that are sent together. They are a *header* and a *ciphertext*. A transmission of these messages is called a *session*. The *ciphertext* is a message encrypted with the *session key* (SK) K_e, while the *header* provides information which is necessary for the privileged nodes to obtain the key K_e. The BC sends a communication package, *msg*,

$$msg = \langle \langle header \rangle, E_{K_e}(M) \rangle, \tag{1}$$

where $E_{K_e}(M)$ denotes a *ciphertext*, that is, the message M encrypted with a symmetric cipher and the key K_e. For the node x, PSK_x denotes a set of preshared keys stored by x. The BE protocol requires some pre-defined function F which should enable the SK recovering only by the privileged nodes:

$$\begin{aligned} &\forall x \in T : F(\langle header \rangle, \mathrm{PSK}_x) = K_e, \\ &\forall x' \notin T : F(\langle header \rangle, \mathrm{PSK}_{x'}) \neq K_e \end{aligned} \tag{2}$$

and such that $F(\langle header \rangle, \mathrm{PSK}_{x'})$ must not provide any information about K_e to the node x' and to an attacker.

In BE schemes, the length of the *header* is a transmission (communication) overhead, the time of $F(\langle header \rangle, \mathrm{PSK}_x)$ calculation is a computation overhead and the size of PSK_x is a memory (storage) overhead in nodes' functioning. In further considerations, we assume that all cryptographic primitives used are secure and all secret keys are sufficiently strong that means that no adversary with limited computational resources is able to break a cryptographic primitive or exhaustively search a secret key space.

Analyzing security of BE, we need basic definitions and terms. Now we define some of them.

Resiliency. To the set S of a BE scheme means that even if an eavesdropper obtained all preshared keys of nodes from some set S then he can obtain no knowledge about a secret common to the member nodes of any other set of nodes T, for the sets such that $S, T \subseteq U, S \cap T = \emptyset$.

The scheme is called k-resilient if it is resilient to any set $S \subseteq U$ of size k.

Managing. It is adding, deleting, and revocation of users; it should be performed without a significant overhead.

Backward Secrecy. It is a property of BE which ensures that newly added users (to the set T) are not able to decrypt previous broadcast content.

Forward Secrecy. It means that when a user is removed from the privileged set T, then he is not able to decrypt later broadcast content.

Traitor Tracing. It is a mechanism for identifying traitor users who gave keys (or decrypted ciphertext) to unprivileged users.

BE systems can be classified as *stateful* and *stateless* schemes. The stateful approach requires that users are always connected to a broadcast center. The BC is used for keys update. Users in stateless schemes cannot update their keys, so a permanent connection with the BC is not required.

In a distributed WSN environment we need an efficient and secure communication protocol. Since nodes are vulnerable to malicious tampering, the BE approach must be resilient or at least k-resilient for high k. The network management should be very efficient from sensor resources point of view. We must note that in a typical network the nodes are not designed for computing but rather for data transmitting [6]. Thus, to decrease energy consumption, the right strategy is to perform *heavy* computations at the BC. Wireless Sensor Networks are dynamic by nature, so the *backward* and *forward secrecy* must be provided by a BE scheme designed for WSNs. Broadcast encryption is also often used in digital rights management (DRM) systems. For these applications (pay TV, DVD protections, etc.), the traitor tracing feature is very helpful to counteract a copyright piracy. This problem has been addressed in past years in many papers, see for example, [11–15]. Full tracing traitors schemes seem to be too exhaustive for WSNs, which are low cost by nature. Nevertheless, in our review, of BE we treat this function as an advantage of the schemes.

2.1. Related Approaches.

The broadcast encryption problem is connected with group key agreement protocols, multicast security, and other constructs designed for secure group

communication. Some of these methods can be used to solve the BE problem in sensor networks.

In literature there are many decentralized schemes for key agreement in distributed sensor networks, see for example, [16–18]. They play a similar role as BE but they work in a quite different way, because these protocols' objective is to agree on a key between two nodes (or, eventually, a larger group of nodes) and to achieve this a Broadcast Center is not deployed. Usually, this class of schemes consists of the following phases: *pre-distribution*, *key discovery* and *path/channel establishment*. *Pre-distribution* is realized during nodes' preparation before the network deployment, but the second and the third phases are realized when the network's nodes are active. Such protocols are ideal for self-organizing applications, but absence of a BC in a key discovery and the path/channel establishment phase may lead to security flaws. In such a case, a malicious attacker can abuse these phases and as a consequence, establish a fake path, revoke some legitimate nodes or perform DoS attack for example, by sending many *discovery messages*. Ramkumar in [19] described a BE scheme based on a random key predistribution. His solution is also decentralized and it does not need the PKC. In further considerations we will concentrate on a group key agreement driven by a broadcast center but the solutions based on the PKC are also noteworthy.

Other efficient constructs in related problems of key distribution are presented in [20]. Besides, introducing a novel authentication scheme, this paper presents a performance comparison of several authentication schemes (including PKC approaches). Brooks et al. in [21] introduced a special infrastructure for security in sensor networks. A network is partitioned into special regions with a chosen node as a key server. Further, within these regions a secure multicast communication is performed.

Papers [22, 23] present surveys of key management schemes in WSNs. The performance of selected schemes is also presented in [24]. A general conclusion from the above reviews is that no key distribution technique is ideal to all scenarios where sensor networks are used. A key distribution protocol selection must depend on a specific network's application, its resources, and characteristics.

2.2. The Broadcast Encryption Schemes. Let us assume for the rest of the paper that $n = |U|$ is a number of users in the whole network and r is a number of users that revoke cooperation. The first formal study of a BE problem was presented in the paper [25] where Fiat and Naor introduced several approaches with different properties. There are schemes that do not require a broadcast center to broadcast messages (they are called *Zero Message Schemes*). These protocols are 1-resilient. Later there were introduced some k-resilient approaches with a low-memory requirement. The most efficient protocol needs storing by each node $O(k \log k \log n)$ preshared keys and broadcasting by a BC $O(k^2 \log^2 k \log n)$ messages. In the paper [26] there is a survey of such related methods. Generally, the review papers focus on comparison of BE schemes in terms of a transmission overhead and a memory overhead. A computational overhead is only slightly remarked. However, in a sensors' case, the computation efforts (and what it follows, energy consumption) is a crucial issue and must be thoroughly analyzed in each solution proposed.

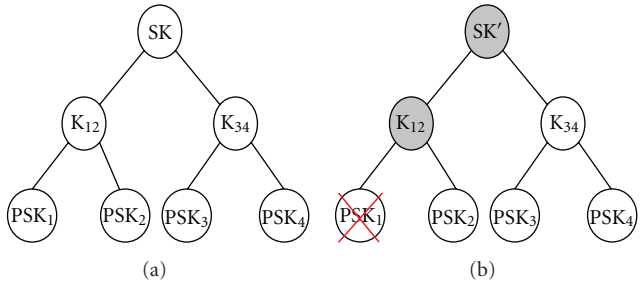

Figure 1: Tree-based broadcast encryption (forming and revocation processes).

The next class of protocols are *tree-based* schemes. This type of secure group communication was proposed independently in two papers. The paper [27] focuses on binary trees, while the paper [28] assumes the degree of the tree as a parameter. The keys are derived by a symmetric cipher and the properties of these protocols are similar. Each user needs to store $O(\log n)$ keys, and the protocol sends $O(\log n)$ (exactly $2 \log n$) messages to establish a new group key after a revocation. An example of the binary tree-based broadcast encryption is showed in Figure 1. Each node has its own PSK and it derives the upper keys with a neighbor using some pairwise key agreement mechanisms. A shared session key (SK) is produced as a root key. The revocation process is presented in the right-hand side of Figure 1. A wrong leaf is deleted, and all keys it possesses should be securely updated. The node with PSK_2 updates K_{12} and SK; next, the nodes with PSK_3 and PSK_4 update only the session key. As we can see, the forming and deleting processes are very important in pairwise key agreement solutions. For such protocols, the communication overhead was reduced in [20, 29]. The improved protocols require only $\log n$ messages to send after a user revocation. Additionally, the methods base mainly on fast cryptographic constructs (hash functions or symmetric ciphers), so they should be applied in networks with limited capabilities. Other well-known, *tree-based* schemes are complete subtree (CS), subset difference (SD) [30], and a layered version of the Subset Difference (LSD) [31]. These schemes are very efficient. The CS method needs $O(\log n)$ PSKs for every node and its transmission cost is $O(r \log(n/r))$ messages sent. In the SD, each node needs to storage $O(\log^2 n)$ PSKs. Next a broadcast center forms a privileged subset by sending $O(r)$ short messages, which mark r revoked nodes. Later, each node has to execute $O(\log n)$ computations. The LSD protocol can achieve the same goal with $O(\log^{1+\epsilon} n)$ PSKs, $O(r)$ messages, and $O(\log n)$ computations.

Some attempts of reducing the nodes' key storage and eliminating the rekeying process are presented in [32]. Another hierarchical scheme that was designed to increase efficiency can be found in [33]. Daza in [34] presented a BE approach for mobile ad-hoc networks. In his method

the transmission overhead is low, but the approach bases on a secret-sharing scheme and the ElGamal-based PKC. The method is interesting but the application of PKC makes it too expensive for standard sensors. In paper [35] Yoo et al., basing on polynomial interpolations, increased the efficiency of the Naor-Pinkas scheme [14]. Their protocol requires $O(\log(n/m))$ PSKs and $O(\alpha r + m)$ messages, where m is the number of partitions of the users set (its choice influence the efficiency of the protocol) and $\alpha \in (1, 2)$ is a predetermined constant. Security of this scheme is provided by the Diffie-Hellman protocol, so deployment of that solution in a WSN may be inefficient.

Sometimes users (nodes of a network) are represented in an alternative way. For example, a novel approach where the set of privileged users is described by attributes, was introduced in [36]. Each node has a set of attributes and a decryption key depends on this set. The advantage of that proposition is that BC can revoke a group of the nodes (not only a single one) sending messages of a size linear with respect to the number of all attributes. It is realized by decryption and restriction of access policy, driven by AND/OR functions on attributes. This solution is resilient and the attributes make it easy to manage, but the challenge is to construct an efficient attribute-based encryption scheme. Another approach is the BE scheme proposed in [37], which operates on users' profiles. It is realized by introducing different types of broadcasts with corresponding probabilities. For each user his profile is created, which denotes probabilities that the given user will subscribe the given broadcast. Next, using profiles, the scheme can optimize some popular tree-based BE protocols. In particular, the solution can reduce a bandwidth required by the complete subtree and by the subset difference approaches. This reduction is significant when the scheme allows some unprivileged nodes to decrypt a content. A tradeoff between storage and communication in BE schemes is an important subject of research. The results of this aspect are presented in papers [38–40]. It is especially important for such environments like a WSN, because of nodes' limitations. A rational tradeoff between a number of preshared keys stored in each node and a volume of broadcast transmission is crucial especially for Wireless Sensor Networks.

3. The One-Time Schemes

The one-time schemes were chronologically the first solutions for the BE problem. We say that a protocol is *one time* if, while using this protocol, PSKs must be updated after every usage. Generally, this is treated as a disadvantage but we will show that the key update after a session period can provide us some additional security benefits. Moreover, such protocols are very efficient from the storage overhead point of view. In majority of *one-time* solutions, a node needs to store $O(1)$ PSKs that is a small number in many scenarios for protocols' efficiency. In Section 4, we will show how to improve the one-time schemes using different PSKs' derivation and authentication methods. Additionally, the one-time schemes can be easily transformed for example, into stateful tree-based schemes. The authentication of nodes is s generic part of BE protocols. Now, we will describe two such schemes of authentication.

Chiou and chen in the paper [41] introduced methods for the BE using a secure lock. Their paper presents public-key and symmetric-key based broadcast protocols. The lock is constructed in such a way, that only a privileged node is able to open it. The construction is based on the Chinese Remainder Theorem (CRT). Each user belonging to T (a privileged node) adds its congruence equation to a set. The system of such equations must be solved by each node to obtain a common secret. The CRT is used as the solution algorithm. Unfortunately, the computational overhead of this scheme is acceptable only for very small groups of nodes, and it is definitely too expensive for sensors with limited capabilities.

Protocols presented by Berkovits in [42] belong to first authentication solutions for BE. The methods are based on two threshold secret sharing schemes: the Shamir's scheme [43] and the Brickell's scheme [44]. We will shortly describe only the first scheme; the second one has similar properties. The Shamir's method uses the polynomial interpolation. Each ith user has the point (x_i, y_i) shared with a Broadcast center as a preshared key. BC carries out the following steps:

(i) it selects the random secret $(0, S)$ and some number (say, j) of additional dummy points (x_i, y_i) with x_i unassigned to a node,

(ii) it finds a polynomial P of degree $k + j$ that passes through the points $(0, S)$ and (x_i, y_i) of the privileged set of k members and through j dummy points and no point of an unprivileged node,

(iii) it broadcasts $k + j$ other points of the graph of P.

By the Lagrange interpolation polynomial, any privileged user is able to calculate $(0, S)$. Next, this secret is used as a session key. The scheme is secure (as the Shamir's scheme is), resilient, and the interpolation can be implemented in a fast way. The BC needs to broadcast $k + j$ messages, each node stores only one PSK. After any change in a privileged set the protocol must be repeated and the PSKs must be updated. In the next section we will show how to enhance that scheme, to make it more secure and flexible.

4. The One-Time Scheme with Additional Capabilities

In this section, at first we will propose methods for keys generation. We will show how a broadcast center and nodes can use a session key and preshared keys to achieve security goals (secret communication). Next we will consider the aspect of the source authentication. We will describe some popular methods realizing this security service that is crucial, especially for WSNs. Adequate keys management and authentication can really improve security and efficiency of BE schemes. The methods that will be proposed in this section are applicable in different schemes but we will show them in case of the Berkovits scheme [42] described in Section 4. Security analysis of the improvements proposed will be presented in Section 5.

Now, let us introduce a notation required in the rest of the paper: $H(\cdot)$ denotes a secure cryptographic hash function, $\mathrm{Mac}_K(\cdot)$ denotes a secure message authentication code (MAC) with the key K, $\|$ is a concatenation of two blocks of bits, $\oplus$ is a XOR (exclusive OR) operation.

In the Berkovits' scheme, a node needs to store one secret point (a point of the polynomial's graph) that is coded as a block of bits. In each session, that point must be updated. During network's operation, one must enumerate the updated session keys. Let's assume that: K_e^i is a session key during ith session, k_x^i is a secret point (coded as a key) assigned to the node x during ith session. Its value is shared only with the BC, s_x is a secret seed of the node x. Its value is shared only with the BC, PSK_x^i denotes the set of preshared keys of the node x in ith session. The node stores only one set of PSKs at the same time (used in an actual session).

4.1. PSK and SK Derivation. As mentioned before, users using a *one-time* scheme must update keys every session. Now we describe this process. Let us define a generic function $\mathcal{G}en(\cdot)$, which is used for updating keys. In order to synchronize the keys $k_x^i = \mathcal{G}en(i, \mathrm{PSK}_x^{i-1})$ is computed by the node x and by the BC. The BC and the nodes must share the same keys to make the steps of the protocol successfully. Additionally, the parties update PSK_x^{i-1} to PSK_x^i. Of course, the BC must generate a new session key (K_e^i) for a new session and create a new *header*. The number of a session (i) is passed by BC in the broadcast message *msg*:

$$msg = \left\langle i, \langle header\rangle, E_{K_e^i}(M)\right\rangle. \tag{3}$$

Now let us introduce two methods of key derivation.

Method I. The first approach is simpler and therefore fast. In this solution, we define a key update as

$$\mathrm{PSK}_x^i = \left\{s_x, k_x^i\right\}, \tag{4a}$$

$$k_x^i = \mathcal{G}en^1\left(i, \left\{s_x, k_x^{i-1}\right\}\right) = H(i\|s_x). \tag{4b}$$

The key derivation is realized by hashing the concatenation of the session number and the secret seed shared among the node x and the BC. We assume that the output length of $H(\cdot)$ is sufficient to produce a secure key. Each node holds only the actual key and the secret seed and it can generate a key for every session very fast.

Method II. The second method is more secure, but sometimes it requires more computations. The key derivation is realized as follows:

$$\mathrm{PSK}_x^i = \left\{k_x^i\right\}, \tag{5a}$$

$$k_x^i = \mathcal{G}en^2\left(i, \left\{k_x^{i-1}\right\}\right) = H\left(k_x^{i-1}\right). \tag{5b}$$

A key for the first session (k_x^0) is preloaded before deployment.

In a new session the previous key k_x^{i-1} is replaced by k_x^i, and we must ensure that after such an exchange the old key k_x^{i-1} is erased from the node's memory. Each node stores only one key.

4.2. Source Authentication. Providing strong authentication in a distributed sensor network is a hard task [45]. To do this one must at first modify the broadcast message *msg* in (3). Now the BC sends

$$msg = \left\langle i, \langle header\rangle, E_{K_e^i}(M), AuthTag\right\rangle. \tag{6}$$

Besides a *header* and an encrypted content, the broadcast message must contain the authentication tag ($AuthTag$). Each legitimate node, using that tag, should be able to check authenticity of the message. Let us denote

$$\begin{aligned} msg &= \left\langle i, \langle header\rangle, E_{K_e^i}(M)\right\rangle, \\ AuthTag &= \mathcal{A}uth(msg). \end{aligned} \tag{7}$$

$\mathcal{A}uth(msg)$ is an authentication function that can be realized in several ways. We will present two popular and one novel method.

The PKC Approach. PKC provides us digital signatures. It is an ideal method to authenticate BE messages, but only when the sender's resources are able to do this. The BC authenticates the broadcast traffic by signing it:

$$AuthTag = \mathcal{A}uth(msg) = Sign(msg). \tag{8}$$

$AuthTag$ is a message signed by the BC and when a user wants to check its authenticity, he uses the function of verification $Verify(msg, AuthTag)$. Such a function is easy to manage and is secure and scalable, but is rather impractical in environments like WSNs. Usually, PKC-based signatures are long and signing/verification operations require a computational overhead exceeding nodes' capabilities. We related to this aspect above, in Section 1.

The Standard Approach. Because PKC in WSNs is not recommended, we must use symmetric methods. Assume that K_a is an authentication key shared between legitimate nodes of a network and the BC. Source authentication is realized by means of MACs as follows:

$$AuthTag = \mathcal{A}uth(\mathrm{msg}) = \mathrm{Mac}_{K_a}(\mathrm{msg}). \tag{9}$$

Verification in this and the next method is a simple computation of the tag $\mathrm{Mac}_{K_a}(\cdots)$ as in (9) and checking if the computed tag and $AuthTag$ (appended with message) are equal. When the authentication key is shared among all members of a group then a sender of a message cannot be identified in a clear-cut way. Note that each owner of K_a can authenticate any message. It is acceptable when a BC has solely an ability to broadcast a content (e.g., in pay TV), but in sensor networks it may cause problems.

The Enhanced Approach. Now we want to improve the standard approach presented above making it useful for WSNs. We propose to attach the list of privileged nodes T to the broadcast message. Additionally, we XOR the session key K_e^i with K_a. Thus, the steps of the authentication protocol are

$$\begin{aligned} msg &= \langle T, i, \langle header\rangle, E_{K_e^i}(M)\rangle, \\ AuthTag &= \mathcal{A}\mathrm{uth}(msg) = \mathrm{Mac}_{K_a\oplus K_e^i}(msg). \end{aligned} \tag{10}$$

Next, the BC distributes the broadcast message *bmsg*:

$$bmsg = \left\langle T, i, \langle header\rangle, E_{K_e^i}(M), AuthTag \right\rangle. \quad (11)$$

These small modifications have some consequences on security and efficiency of the protocol. We will present them in Section 5.

5. Analysis

Now, we will analyze the foregoing methods. We will start from analyzing functions of the BE scheme, next we will focus on the rekeying process and the authentication approaches while an analysis of security and performance will summarize this section.

The BE Functions. A *newcomer* is a node which is new in a privileged set in an actual session. j is the number of newcomers, and i is the number of an actual session. When we want *add* nodes to a privileged set, a natural way is to create a new set T, a new SK and a new *header*, and next broadcast them. In our case, this needs sending $O(|T|)$ messages, performing $O(1)$ computations in each node (using the key derivation in (5b) requires $O(i)$ computations for newcomers). Such a solution holds the *backward secrecy,* but the communication overhead is significant.

The easiest way is sending unicast to newcomers an encrypted message containing the session number i and the key k_e^i. It requires only $O(j)$ messages and no computations is required. However, this solution contradicts the *backward secrecy* and new nodes can decrypt all traffic along the session i. We must ensure that the newcomers are not able to decrypt previous messages. It can be achieved by sending only one additional message and performing one operation in the privileged nodes. During the session i the BC derives the new key $K_e^{i+1} = H(K_e^i)$, next it sends the key K_e^{i+1} to the newcomers and broadcasts an *update* message, which denotes that any privileged node should perform the calculation $K_e^{i+1} = H(K_e^i)$. The key K_e^i should be erased from the memory. Now, the session is updated to $i + 1$ and the broadcast traffic is encrypted with the key K_e^{i+1}. This is an efficient method. It requires only $O(j)$ short unicast messages and $O(1)$ computations in each node. The main advantage of that approach is assurance of the *backward secrecy*. The newcomers in the session $i + 1$ have the key K_e^{i+1} and they are not able to compute the previous key K_e^i.

Efficient *deletion* of nodes from T is one of the hardest tasks in BE protocols. Nodes leave, fail and sometimes we want to revoke malicious nodes. Presented BE scheme does not provide an efficient deletion function. If we want to delete some nodes then the BC must make a new session without these nodes. When we want to delete some nodes the BC must make a new session without these nodes. In that operation the *forward secrecy* is ensured. The BC generates a new SK independently, so the revoked node cannot decrypt present and future messages. Such an operation requires $O(|T|)$ messages and $O(1)$ operations in nodes.

An interesting solution for improving the performance of revocation is forming subsets of privileged nodes. We can divide T into several subsets and perform a BE scheme on these subsets separately. This way we achieve subkeys and now we can repeat the process until we will generate a common SK. That strategy is related to *tree-based* schemes that are described in Section 2, or to other hierarchical constructions. Such a method can be used for pairwise key derivation in tree-based settings. Thus, the overheads are similar to overheads in other tree-based solutions presented in Section 2.

A *traitors-tracing* service known from DRM systems is not available, but the scheme allows to trace the source of a preshared key leak. When a pirate node uses a passed SK to decrypt the content, we are not able to determine the source of the leak. However, the SK for any session is different, so a traitor must pass a SK in each session. In a WSN it may be discovered by an anomaly detection or an intrusion detection system. More serious is an adversary's active attack. A privileged node can be captured and its session can be cloned into other nodes. Now an attacker is able to decrypt the traffic. When we detect a piracy hardware and tamper it, we can determine which node was captured. The BC keeps all seeds and actual keys of the node what requires $O(n)$ keys at the BC. When a protocol uses the method of key derivation proposed in (4b) then the BC needs to perform $O(n)$ operations, while using the tracing given in (5b) requires $O(in)$ computations. This second effort ($O(in)$) can be reduced using the method presented in [46] to $O(n\log^2 i)$.

Rekeying versus Not ReKeying. BE schemes are designed to hold *backward secrecy* and *forward secrecy* properties in a sense of protection of unassigned messages against users joining or leaving the privileged set. However, in WSN-like applications, we should be more demanding. Consider a situation presented previously when an adversary captures a privileged node. It is standard assumption in environments like sensor networks. If a BE protocol does not require rekeying in each session, then an adversary having PSKs of a privileged node and its previous traffic can decrypt it all. In networks, where an adversary has capabilities to compromise nodes, we need *backward secrecy* assurance. To achieve this property, we must deploy some method of rekeying. We presented two approaches: in (4b) and in (5b). By solution given in (4b), we can generate a key of any session and at any moment. It is fast, but after compromising PSKs, also an adversary is able to obtain a key of any session. The approach given in (5b) holds *backward secrecy*. An actual user's key is produced from the previous key $k_x^i = H(k_x^{i-1})$. An adversary capturing a node has only one key which is the actual key. He can decrypt present and future messages, but he has no way to achieve previous keys. The only disadvantage is that a node which joins the privileged set must synchronize its key. In the extreme case a computational overhead of such a synchronization is $O(i)$. How to improve it by unicast messages we already described in this section.

Authentication. In this paragraph of the paper, we will present an analysis of authentication schemes presented in Section 4. We omit authentication schemes based on PKC due to reasons mentioned throughout the paper and we will focus on symmetric methods of authentication that are very efficient in WSNs, see for example, [47, 48]. At first we consider standard approach from (9). It is a good method

when authentication is realized between two parties. When a large group shares the same authentication key, each member can send or modify a message and the authentication will pass. We can improve (9) by a concatenation of the key K_e^i to an input of the MAC function: $\mathrm{Mac}_{K_a}(K_e^i \| msg)$. Nevertheless, that approach is still insecure. First, when verification process fails, a node is not sure if a message is false or if it is outside of a privileged set (it is not able to achieve the key K_e^i). Next, an adversary with a compromised privileged node can disrupt sessions, each message can be authenticated by him. He can jam the original broadcast data and can create fake sessions with higher session numbers, in order to force nodes to synchronize session numbers and session keys. Such an attack can be destructive, when nodes derive keys using the method given in (5b). An adversary can also try to execute a DoS attack. Nodes, to verify *AuthTag*, must compute the key K_e^i. The adversary can just send (many times) big instance of BE problems and the nodes will be exhausted by computing them. These disadvantages are generic if a large group shares one authentication key.

Now, we will analyze next authentication method given in (10), which improves the previous one. The BC broadcasts the following message *mes*:

$$\begin{aligned} AuthTag &= \mathrm{Mac}_{K_a \oplus K_e^i}\left\langle T, i, \langle header\rangle, E_{K_e^i}(M)\right\rangle, \\ mes &= \left\langle T, i, \langle header\rangle, E_{K_e^i}(M), AuthTag\right\rangle. \end{aligned} \tag{12}$$

Sending the privileged set T as a list of receivers T is an additional communication overhead. However, consider random nodes' enumeration from the set $0, \ldots, n$. Only a node (and the BC) knows its own number. Then, we can encode the set T as a $\log_2|T|$-bits long binary mask. 0s and 1s on corresponding places in the mask denote that 0-marked nodes are unprivileged and 1-marked ones are privileged. Thus, even in large networks that overhead is acceptable. The solution presented influences efficiency. A node before any processing only checks if its bit is 1, otherwise it discards the message. In the previous methods, nodes always tried to achieve the SK. That improvement saves energy and provides other capabilities to the network (e.g., a possibility of routing). However, the main goal is the authentication. The BC, by sending list of T members, declares for which nodes the message is destined. This declaration means that any privileged node, after obtaining the key K_e^i from the *header*, is able to verify the signature *AuthTag*. If the verification fails then this means that the message is fake or a transmission error occurred. In both cases the node should send an alarm message. Consider now an adversary owning a group of nodes. He can create a message like in (12), but he must declare only the captured nodes in the list T of privileged nodes. Thus, any honest node even does not start processing a malicious message because it is not on the list of receivers attached by the adversary. If the adversary adds to that list a node, which is not compromised, then the node will start obtaining the key K_e^i and the signature's *AuthTag* verification will fail. This is because the adversary does not know the node's PSKs and he is not able to create a correct *header* without such a knowledge. An uncompromised node should alarm the network. This way we achieved such useful properties of the network as

(i) authentication of a fake message will pass if and only if the list of receiver nodes contains only compromised nodes,

(ii) if an adversary adds an uncompromised node to the receivers' list then that added node is able to detect the forgery.

This means that a source which broadcasts a content must know all PSKs of the set of nodes T he declared, otherwise a regular node from that list can detect a forgery. However, the node processes a message only if it is on declared in the list, so a fake messages will not desynchronize a session. These features make our authentication scheme very useful in broadcast communication.

Security and Performance. Now we consider security of the scheme given in (10).

> Security of Scheme based on Shamir's Secret Sharing Scheme which is *perfect secure* (information-theoretic secure) [42, 43].
>
> Construction is resilient [42], so an attacker cannot obtain a session key.
>
> Now we should consider confidentiality and integrity of the scheme.
>
> Assume that encryption $E_K(\cdot)$ is indistinguishable under chosen plaintext attack (IND-CPA secure) and MAC scheme $\mathrm{Mac}_K(\cdot)$ is strongly unforgeable under chosen-message attack (SUF-CMA secure).
>
> Then, Theorems 4.4 and 3.2 from the paper [49] imply that construction from (10) is IND-CCA secure.

Now, we evaluate performance of our scheme. We assume that we use the presented scheme in tree-based setting (Figure 1), and we compare it also with the stateful tree-based broadcast encryption. We also assume that its security level is 128 bits. According to the properties of the schemes, their transmission and storage overheads are the same. The only difference is a computational overhead in the pairwise keys agreement process. The standard solutions use cryptographic primitives, thus for tests we chose assembler implementation [48] of the AES [50] block cipher. The polynomial interpolation required was implemented in C. ATmega1281 Microcontroller (with 8 MHz clock speed, 8 KB of RAM and 128 KB of Flash) was selected as a characteristic platform for a regular node in the sensor network. One pairwise rekeying function takes 0.4 ms on each sensor using the block cipher. At the same platform, the polynomial interpolation takes less than 0.1 ms. The implementation of the polynomial interpolation needs only 1329 bytes, while the AES uses 2141 bytes of storage.

6. Conclusions and Future Research

In this paper, we considered the BE problem in distributed wireless sensor networks. We defined the problem, described

main existing solutions, and next focused on *one-time schemes*, a type of schemes that are most suitable for WSNs. The one-time schemes are resilient, that is essential in such applications. Such protocols are very attractive for WSNs also in other respects. Their storage overhead of the range $O(1)$ (exactly up to 2 PSKs) is minimized as possible. We shown how standard BE operations may be realized to achieve efficient protocols, that is, with $O(1)$ operations and $O(j)$ messages in *add* operation and with $O(1)$ computations and $O(|T|)$ transmissions for *delete* operation. To improve transmission and revocation overheads, the protocols can be easily optimized, for example, by applying a tree-based structure. The schemes satisfy the *backward secrecy* and the *forward secrecy* properties. The *one-time* protocols are criticized due to a requirement of rekeying in each session. In our paper we showed that, unexpectedly, the *one-time* schemes in some applications are much better than other protocols. Because of the rekeying process in each session we can achieve the *backward secrecy* of PSKs. Another contribution to BE is including the authentication service to the protocols. We presented a symmetric cryptography-based method which has very useful security features. The scheme is also very efficient. The polynomial interpolation used is about four times faster than a fast block cipher encryption. This method ideally fits to WSNs and ensures that only a BC can generate authenticated sessions.

In this paper we extended the specific Berkovits' scheme [42], but we did not stress this fact, because our modifications are mainly generic and can be adopted in majority of BE schemes. The scheme used is relatively fast but in future research we will focus on designing more efficient *one-time* BE schemes. The new constructions will be dedicated to sensor networks. Another interesting subject of research is the *freshness* aspect in BE. It ensures that data transmitted is fresh beside of being confidential and authentic. Compiling the security services of data confidentiality, authenticity and freshness will make BE protocols more secure remaining them lightweight.

Acknowledgment

This work is partially supported by the National Science Center (NCN), under Grant with decision's number DEC-2011/01/N/ST7/02995 and by the 7FP NoE EuroNF project.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.

[2] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[3] N. Xu, "A survey of sensor network applications," http://courses.cs.tamu.edu/rabi/cpsc617/resources/sensor%20nw-survey.pdf.

[4] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2000.

[5] P. Szalachowski, Z. Kotulski, and B. Ksiezopolski, "Secure position-based selecting scheme for wsn communication," in *Computer Networks*, A. Kwiecień, P. Gaj, and P. Stera, Eds., vol. 160 of *Communications in Computer and Information Science Computer Networks*, pp. 386–397, Springer, Heidelberg, Germany, 2011.

[6] D. W. Carman, P. S. Kruus, and B. J. Matt, "Constraints and approaches for distributed sensor network security," Tech. Rep. 010, NAI Labs, The Security Research Division Network Associates, 2000.

[7] M. Brown, D. Cheung, M. Kirkup, and A. Menezes, "Pgp in constrained wireless devices," in *Proceedings of the 9th USENIX Security Symposium*, pp. 247–261, USENIX, Denver, Colo, USA, August 2000.

[8] A. S. Wandert, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom '05)*, pp. 324–328, March 2005.

[9] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, 2004.

[10] T. Kavitha and D. Sridharan, "Security vulnerabilities in wireless sensor networks: a survey," *Journal of Information Assurance and Security*, vol. 5, no. 1, pp. 31–44, 2010.

[11] B. Chor, A. Fiat, and M. Naor, "Tracing traitors," in *Proceedings of the 14th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '94)*, pp. 257–270, Springer-Verlag, London, UK, 1994.

[12] D. Boneh and M. K. Franklin, "An efficient public key traitor tracing scheme," in *Proceedings of the 19th Annual International Cryptology Con- ference on Advances in Cryptology (CRYPTO '99)*, pp. 338–353, Springer-Verlag, London, UK, 1999.

[13] E. Gafni, J. Staddon, and Y. L. Yin, "Efficient methods for integrating traceability and broadcast encryption," in *Proceedings of the 19th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '99)*, pp. 372–387, Springer-Verlag, London, UK, 1999.

[14] M. Naor and B. Pinkas, "Efficient trace and revoke schemes," in *Proceedings of the 4th International Conference on Financial Cryptography (FC '00)*, pp. 1–20, Springer-Verlag, London, UK, 2001.

[15] D. Naor, M. Naor, and J. B. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '01)*, pp. 41–62, Springer-Verlag, London, UK, 2001.

[16] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS '02)*, pp. 41–47, Association for Computing Machinery, New York, NY, USA, November 2002.

[17] R. di Pietro, L. V. Mancini, and A. Mei, "Random key-assignment for secure wireless sensor networks," in *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor networks (SASN '03)*, pp. 62–71, Association for Computing Machinery, New York, NY, USA, October 2003.

[18] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Proceedings of the 2003 IEEE Symposium on Security and Privacy (SP '03)*, p. 197, IEEE Computer Society, Washington, DC, USA, 2003.

[19] M. Ramkumar, "Broadcast encryption with random key pre-distribution schemes," in *Proceedings of the 1st International*

Conference on Information Systems Security (ICISS '05), Kolkata, India, 2005.

[20] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast security: a taxonomy and some efficient constructions," in *Proceedings of the 18th Annual Joint Conference of the IEEE Computer and Communications Societie (INFOCOM-99)*, pp. 708–716, March 1999.

[21] R. R. Brooks, B. Pillai, M. Pirretti, and M. C. Weigle, "Multicast encryption infrastructure for security in sensor networks," *International Journal of Distributed Sensor Networks*, vol. 5, no. 2, pp. 139–157, 2009.

[22] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway, "A survey of key management schemes in wireless sensor networks," *Computer Communications*, vol. 30, no. 11-12, pp. 2314–2341, 2007.

[23] S. A. Çamtepe, B. Yener, and M. Yung, "Expander graph based key distribution mechanisms in wireless sensor networks," in *Proceedings of the 2006 IEEE International Conference on Communications (ICC '06)*, pp. 2262–2267, Istanbul, Turkey, July 2006.

[24] Y. Amir, Y. Kim, C. Nita-Rotaru, and G. Tsudik, "On the performance of group key agreement protocols," *ACM Transactions on Information and System Security*, vol. 7, no. 3, pp. 457–488, 2004.

[25] A. Fiat and M. Naor, "Broadcast encryption," in *Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, pp. 480–491, Springer-Verlag, York, NY, USA, 1994.

[26] J. Horwitz, "A survey of broadcast encryption," Tech. Rep., 2003.

[27] D. Wallner, E. Harder, and R. Agee, "Key management for multicast: issues and architectures," 1999.

[28] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16–30, 2000.

[29] D. A. McGrew and A. T. Sherman, "Key establishment in large dynamic groups using one-way function trees," *IEEE Transactions on Software Engineering*, vol. 29, no. 5, pp. 444–458, 2003.

[30] D. Naor, M. Naor, and J. Lotspiech, "Revocation and tracing schemes for stateless receivers," in *Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology*, pp. 41–62, Springer-Verlag, Santa Barbara, Calif, USA, 2001.

[31] D. Halevy and A. Shamir, "The lsd broadcast encryption scheme," in *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '02)*, pp. 47–60, Springer-Verlag, Santa Barbara, Calif, USA, 2002.

[32] C. Chang, Y. Su, and I. Lin, "A broadcast-encryption-based key management scheme for dynamic multicast communications work-in-progress," in *Proceedings of the 2nd International Conference on Scalable Information Systems (InfoScale '07)*, pp. 69:1–69:2, Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Suzhou, China, 2007.

[33] A. D. Santis, A. L. Ferrara, and B. Masucci, "Efficient provably-secure hierarchical key assignment schemes," in *Proceedings of the 32nd International Symposium on Mathematical Foundations of Computer Science (MFCS '07)*, Cesky Krumlov, Czech, August 2006.

[34] V. Daza, J. Herranz, P. Morillo, and C. Ràfols, "Ad-hoc threshold broadcast encryption with shorter ciphertexts," *Electronic Notes in Theoretical Computer Science*, vol. 192, no. 2, pp. 3–15, 2008.

[35] E. S. Yoo, N. S. Jho, J. H. Cheon, and M. H. Kim, "Efficient broadcast encryption using multiple interpolation methods," *Lecture Notes in Computer Science*, vol. 3506, pp. 87–103, 2005.

[36] D. Lubicz and T. Sirvent, "Attribute-based broadcast encryption scheme made efficient," in *Proceedings of the Cryptology in Africa 1st international conference on Progress in cryptology (AFRICACRYPT '08)*, pp. 325–342, Springer-Verlag, Casablanca, Morocco, 2008.

[37] M. Ak, K. Kaya, K. Onarlioglu, and A. A. Selçuk, "Efficient broadcast encryption with user profiles," *Information Sciences*, vol. 180, no. 6, pp. 1060–1072, 2010.

[38] C. Blundo, L. A. F. Mattos, and D. R. Stinson, "Trade-offs between communication and storage in unconditionally secure schemes for broadcast encryption and interactive key distribution," in *Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology (CRYPTO '96)*, pp. 387–400, Springer-Verlag, Santa Barbara, Calif, USA, 1996.

[39] R. Canetti, T. Malkin, and K. Nissim, "Efficient communication-storage tradeoffs for multicast encryption," in *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques (EURO-CRYPT '99)*, pp. 459–474, Springer-Verlag, Prague, Czech, 1999.

[40] C. Padró, I. Gracia, and S. Martín, "Improving the trade-off between storage and communication in broadcast encryption schemes," *Discrete Applied Mathematics*, vol. 143, no. 1–3, pp. 213–220, 2004.

[41] G. H. Chiou and W. T. Chen, "Secure broadcasting using the secure lock," *IEEE Transactions on Software Engineering*, vol. 15, no. 8, pp. 929–934, 1989.

[42] S. Berkovits, "How to broadcast a secret," in *Theory and Application of Cryptographic Techniques*, vol. 547, pp. 535–541, 1991.

[43] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[44] E. F. Brickell, "Some ideal secret sharing schemes," in *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, pp. 468–475, Springer-Verlag, New York, NY, USA, 1990.

[45] B. Lampson, M. Abadi, M. Burrows, and E. Wobber, "Authentication in distributed systems: theory and practice," *ACM Transactions on Computer Systems*, vol. 10, no. 4, pp. 265–310, 1992.

[46] D. Coppersmith and M. Jakobsson, "Almost optimal hash sequence traversal," in *Proceedings of the 6th international conference on Financial cryptography (FC '02)*, pp. 102–119, Springer-Verlag, Berlin, Germany, 2003.

[47] P. Szalachowski, B. Ksiezopolski, and Z. Kotulski, "On authentication method impact upon data sampling delay in wireless sensor networks," in *Computer Networks*, A. Kwiecień, P. Gaj, and P. Stera, Eds., vol. 79 of *Communications in Computer and Information Science*, pp. 280–289, Springer, Berlin, Germany, 2010.

[48] P. Szalachowski, B. Ksiezopolski, and Z. Kotulski, "CMAC, CCM and GCM/GMAC: advanced modes of operation of symmetric block ciphers in wireless sensor networks," *Information Processing Letters*, vol. 110, no. 7, pp. 247–251, 2010.

[49] M. Bellare and C. Namprempre, "Authenticated encryption: relations among notions and analysis of the generic composition paradigm," *Journal of Cryptology*, vol. 21, no. 4, pp. 469–491, 2008.

[50] J. Daemen and V. Rijmen, *The Design of Rijndael*, Springer-Verlag, Secaucus, NJ, USA, 2002.