Key Management for Security in Wireless Networks

Urvashi Sangwan

Assistant Professor, Vaish College of Engg, Rohtak,
Haryana-124001, India

Abstract

Wireless Sensor Network (WSNs) have a vast field of applications, including environment monitoring, battlefield surveillance and targeting system. As WSNs are usually deployed in remote or even hostile environments and sensor node are prone to node compromise attacks, the adoption of dynamic key management is extremely important. However, there source-constrained nature of sensor nodes hinders the use of dynamic key management solutions designed for wired and adhoc networks. Hence, many dynamic key management schemes have been proposed for WSNs recently. This paper investigates the special requirements of dynamic key management in sensor network environment, and introduces several basic evaluation metrics. In this work, the state of the art dynamic key management schemes are classified into different groups and summarized based on the evaluation metrics. Finally, several possible future research directions for dynamic key management are provided.

Keywords: *cryptography, sensor node, WSN, key management.*

1. Introduction

A wireless sensor network (WSN) consists of a large number of sensor nodes, which are powered by batteries, equipped with sensing, data processing and short-range radio communication components. The applications of WSNs range from the most popular ones, like environment monitoring and home automation, to more demanding ones in military or security areas, like battle field surveillance, targeting and target tracking systems. However, the wireless connectivity, the close interaction among sensor nodes and their unattended operation, as well as the absence of physical protection make WSNs vulnerable to a wide range of network-level attacks and even physical.

1.1 Key management primitive

Key is a piece of input information for cryptography algorithms. First, if the key is disposed, the encrypted information would be disclosed. The secrecy of the symmetric key and private key must be assured locally. The Key Encryption Key (KEK) approach could be used at local hosts. Second, key distribution and key agreement over an insecure channel is at high risk and suffers from potential attacks. In the traditional digital envelop approach, a session key is generated at one side and is encrypted by the public-key algorithm, then it is delivered and recovered at other end. In the Diffie-Hellman (DH) scheme, communication parties of both sides exchange some public information and generate a session key on both ends. However, in mobile ad hoc networks the computation load and complexity of key agreement protocol is strongly restricted by node's available resource, dynamic network topology, or network synchronization difficulty. Third, key integrity and ownership should be protected from advanced key attacks.

1.2 Trust models

The authentication of key ownership is the first step for secure communication. Otherwise it is easy to forge or spoof someone's key. So certain trust framework must present to verify the key ownership. For PKI in the public key cryptosystem, there are two dominate trust models, namely, centralized trust model and the web-of-trust trust model. For network scalability the centralized trust mode could be in a hierarchical trust structure instead of a single CA entity. Multiple CA roots could be necessary for a large network, like Internet. There are two major variations proposed in ad hoc network, which we named CA-view trust model and hybrid trust model. The hybrid mode is to glue the centralized and the distributed trust system together. See Figures 1 (a) to 1 (d) for different trust models. In the figures, all nodes within the circle consists a network domain. In Figure 1 (a), there is one entity (in black) that is trusted by all nodes within the domain. In Figure 1 (b), there is no well trusted entity by all hosts in network domain; instead peer node trusts each other and produces the "certificate" based on local trust.

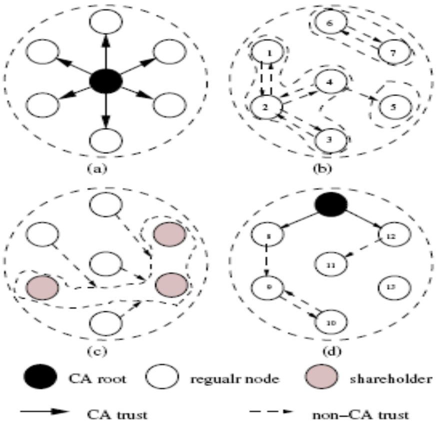


Fig. 1. Different trust models

Figure 1 (c) however shows that quorum nodes (in grey) collaboratively create a view of CA, who functions as CA within the domain. The quorum nodes jointly produce the certificate. Figure 1 (d) shows a combination of (a) and (b) where some nodes are certified by central CA (in black); some are certified by peer nodes. For example, node8and node12 are CA certified, node 9 is not certified by CA but by node 8. Node 13 is not trusted by any node within the domain. Confidence value of CA trust is higher than the value of the peer trust. For example, the value of solid trust line is higher than the dashed line. Each trust line could have different value. Of course, this hybrid trust mode could have further variations. For example, the central CA could be distributed to quorum of nodes. Obviously, in mobile ad hoc networks, framework of key management which built on a fully centralized mode is not feasible not only because of the difficulty to maintain such a globally trusted entity but also the central entity could become a hot spot of attacking, thus network suffers from the security bottleneck. Meanwhile a completely distributed model may not be acceptable because of no well-trusted security anchor available in the whole system. One feasible solution is to distribute the central trust to multiple or entire network entities based on secret sharing scheme. In SEKM, the system public key is distributed to whole network, while the system private key is split to all server nodes.

2. Using network-wide keys

The most straightforward key distribution possible is to have a single master key which is loaded into all sensors. Such simplicity results in a high level of efficiency and flexibility, requiring minimal memory for the storage of keys no matter the size of the network. By loading the master key in new nodes, the scheme also allows the introduction of any number of sensors after the initial deployment. Furthermore, since all nodes certainly share the same master key, this scheme provides perfect key connectivity. A simple scheme that adopts a single secret network-wide key for its operation is the BRO ad cast Session Key Negotiation Protocol (BROSK) [17]. In this solution, the master key K is used in combination with random nodes NA and N B, exchanged by pairs of nodes A and B, for establishing a session key
KA, B= PRF (K||NA||NB), where PRF is a Pseudo Random Function.

3. The full pair wise scheme

While the previous schemes used a single key for the communication between all sensors, the Full Pair wise scheme adopts the extreme opposite approach. In this case, each of the nodes in the network receives 1 pair wise keys to communicate with every other node. This approach assures a high security level, providing features such as mode-to-node authentication and perfect resilience, which thwarts node replication attacks. It also makes the revocation of individual sensor nodes easier: even without the intervention of a secure base station, the nodes on the net-work may identify malicious IDs and revoke the corresponding pair wise keys, e.g., by using voting schemes. The main drawback of this solution is the great memory overhead it introduces, since each node have to store many keys (and many of them may never be used). Due to the lack of resources in the sensors, this is a shortcoming that can greatly limit the scheme's applicability. Moreover, the introduction of new nodes in the network would only be possible if their keys were already loaded from the beginning, which becomes a serious restriction when the net-work needs to be expanded over the initial expectations. Due to these flexibility issues, the Full Pair wise Key scheme could be effectively used basically in small networks where the maximum number of nodes can be predicted with good reliability.

4. Probabilistic approaches

In probabilistic schemes, each node receives a group of keys, the so-called key chain, whose size is normally much lower than the size of the network itself. The reasoning behind this strategy is to provide a fairly good key connectivity and, at the same time, avoid both the memory overhead involved in the Full Pair wise scheme and the low security level offered by a single master key. In general, it is possible to identify three distinct and sequential phases on such schemes, which also appear in other schemes that do not provide perfect connectivity:
1. Key pre-distribution: In this initialization phase, the Key Distribution Center (KDC) chooses each sensor's keychain from a large pool of keys . These chains are then loaded

into the sensors prior to deployment. Each key in the pool usually receives a unique ID, used by the net-work for its identification.

2. Shared-key discovery: After deployment, the sensor nodes try to discover who their neighbours are and which keys they have in common. This phase can be performed either proactively (i.e., neighbouring nodes try to establish keys even before they need to communicate) or in a reactive manner (i.e., shared keys are established on demand). Whenever two nodes establish a shared key, we say that there is a direct link between them. We note that the number of neighbours found in this manner can be increased if the nodes temporarily raise their radio transmission range, as pro-posed in.

3. Path-key establishment: Whenever the key management scheme employed does not provide perfect key-connectivity, some neighbouring nodes may not have keys in common. Thus, if nodes A and B need to establish a secure communication, they have to find a intermediary node C that shares a common key with both A and B .Node C can then act as a mediator for the messages exchanged between A and B or, in order to avoid this extra communication overhead, C can create and distribute a new key to be used by A and B. In either case, we say that an indirect link exists between A and B; however, the revocation mechanisms for the keys generated in the second case may become more complex than those used for pre-existent keys.

4.1. Random key pre-distribution (Basic Scheme)

The Random Key Pre-distribution scheme [13]is considered by many authors as the Basic Scheme. During its Key Pre-Distribution phase, a large key pool P is initialized with |P| random keys and their respective identifiers. For each node, k keys are drawn at random from P. These keys are then loaded into the node's memory, forming its keychain. Using the theory of random graphs [26, Section1.1.1], the exact values of |P| and k can be chosen in such a manner that each pair of nodes share at least one key with an arbitrary probability.

During the Shared-key Discovery phase, each node broadcasts a list containing the IDs of all keys in its chain, allowing neighbouring nodes to identify which keys they have in common.

Variants of this approach adopting a challenge-response technique could also be used to improve the security of this phase. For example, node A could send a message of the form

{α, Eke(α), i=1, 2,…., k}

Where α is a challenge; the correct decryption of a by a node B receiving this message would allow B to discover the shared keys [13]. The disadvantage of this strategy, which is similar to Markel Puzzles [23], is the greater communication and processing overheads it introduces. In the Path-key Establishment phase, any pair of nodes A and B having no key in common must find an intermediary node C. Any node whose key chain contains key IDs present in both A's and B's chain is a suitable candidate. Upon request, C can choose unassigned keys from its keychain in order to create an indirect link between A and B. A possible extension of this scheme for providing revocation capabilities deploys controller nodes, trusted entities having a large communication range. During the pre-distribution phase, these especial nodes receive the identification of some sensors, the IDs of the keys in their corresponding key chains and a shared key with all sensors in the network. Whenever a compromised node needs to be revoked, the controller broadcasts a single revocation message enclosing a signed list of the k key IDs pertaining to the revoked node's key chain. The list signatures are generated using the keys shared between the controller node and the other sensors, and it is sent in uncast manner. After receiving and verifying the list signature, the nodes remove the corresponding keys from their memory and, if needed, perform the Shared-key Discovery and Path-key Establishment phases in order to recover broken links. The Basic Scheme is fairly simple, but it is interesting to provide a connected network with a reduced amount of memory for storing keys. The scalability and the resilience of the scheme are highly dependent on the sizes of the key pool and key chains. Moreover, the existence of trusted controller nodes is not common in all applications (in fact, they are more common in heterogeneous networks), making the revocation of compromised nodes a difficult issue. Furthermore, this solution has some disadvantages such as the lack of node-to-node authentication features and the considerably high communication overhead. Thus, the Basic Scheme's importance resides mainly in the fact that many subsequent key management proposals have been developed aiming at overcoming its limitations.

4.2. Cluster key grouping

The Cluster Key Grouping [16] scheme proposes a modification to the Basic Scheme where the key chains are divided into clusters. Each cluster receives a start key ID, which implicitly determines all other IDs in that cluster. With this strategy, the messages broadcast during the Shared-key Discovery phase can carry only c start key IDs, while the Basic Scheme would require a total of k>=c IDs. Hence, the adoption of large clusters results in the broadcast of few IDs. However, the size of the clusters must be chosen carefully for achieving a same key-connectivity, networks with larger clusters require their nodes to store a larger number of keys. Therefore, the Cluster Key Grouping provides an interesting trade-off between communication and memory-efficiency to the

Basic Scheme, while keeping the flexibility and security properties of the latter.

4.3. Hashed random key pre-distribution

Another simple modification to the Basic Scheme is to hash the keys from the key pool a different number of times for distinct nodes, as proposed in the Hashed Random Key Pre-distribution (RKP-H) scheme [25]. In this solution, only the first node getting the key Ki from the pool receives it as is, while the j th node receive its (j - 1)-times hashed version Hashj-1(Ki), as well as the value of j. During the Shared-key Discovery phase, nodes A and B inform not only the key IDs, but also the value of j for each of them; in this manner, if nodes A and B are loaded, respectively, with
KA = Hash ja (Ki) and KB = Hash jb (Ki) (ja> jb),
then B can easily compute
KA = Hash ja-jb (KB)
The net result of this modification is that the capture of node C and of its key
KC= Hash jc(Ki), will compromise only the keys
KD = Hash jd(Ki) for which jd> jc; in comparison, the capture of node C in the Basic Scheme would reveal Ki itself, compromising all nodes that received that key. Therefore, the RKP-H scheme trades some storage, communication and computation overhead for extra resilience, keeping the remaining properties of the Basic Scheme.

4.4. The Q-Composite scheme

Chan et al. [8] proposed a modification on the Basic Scheme aiming to increase the network resilience at the cost of some processing overhead. In this solution, denominated Q-Composite scheme, two nodes can establish a direct link only if they have at least q > 1 keys in common, instead of single one. Hence, after the key-discovery phase, the key effectively used to encrypt the link between two nodes A and B is computed as
KA;B=Hash (K1‖K2‖ _ _ _ ‖Kq'),
where q' >= q stands for the actual number of shared keys between the nodes. So, as q increases, it becomes exponentially harder for an attacker to recover all the keys needed to break a link. However, for a given network connectivity, the size of the key pool in the Q-Composite scheme is smaller than in the Basic Scheme, thus allowing attackers to recover a larger portion of the network keys by capturing fewer nodes. The combination of these two factors results in a solution that, in comparison with the Basic Scheme, isomer resilient when few nodes are captured, but becomes less secure when many nodes are captured. This may actually be an attractive trade-off in many applications because small scale attacks are expected to be cheaper to mount and harder to detect than large-scale attacks.

4.5. Multipath key reinforcement

The motivation behind the Multipath Key Reinforcement [8] resides in the fact that, after the completion of the Shared-key Discovery phase in solutions such as the Basic Scheme, many direct links are protected by a same key Ki, which may be known by many nodes in the network. Thus,2596 M.A. Implicit Jr. et al. / Computer Networks 54 (2010) 2591–2612 the capture of a single node A having Ki in its key chain will compromise all those links. The Multipath Key Reinforcement is a proposal to strengthen the security of established links, using techniques also explored in [1]. The basic idea of the scheme is to update the link keys for nodes A and B after the Shared-key Discovery phase. This update is performed through multiple disjoint paths, i.e., paths that do not have physical links in common. Hence, depending on the requirements of the target deployment scenario, this trade-off between security and efficiency may hinder the applicability of the Multipath Key Reinforcement.

4.6. Session key scheme

The Session Key Scheme [15] provides a way to create session keys for each interaction between nodes. In this proposal, the Key Pre-Distribution and Shared-key Discovery phases proceed exactly as in the Basic Scheme combined with the Multipath Key Reinforcement. However, the key KA,B established between nodes A and B is not used directly for encrypting their communication. Instead, it issued as the initial key for computing the session key Ki = -Hash i(Ki-1,KA,B) (i > 0), where K0 is a publicly known seed. The exact value of i for each communication session is taken from an agreed sequence I = {i1, . . ., is}, which is computed in the following (unencrypted) manner: m arrays containing s random numbers are sent from A to B via different paths; I is obtained as the result of XO Ring these arrays together and then sorting the s values in ascending order.

The adoption of different keys for each session makes the Session Key Scheme more secure than the simple combination of the Basic Scheme and the Multipath Key Reinforcement. However, this strategy has a very limited effect over the network's resilience, since an attacker that is able to recover the initial keys needs only to eaves drop the distribution of the arrays that form I to compute the correct session keys. Therefore, this slight security boost may not compensate the added computation and communication overheads.

4.7. Key redistribution scheme

Law et al. [18] proposed a modification of the Basic Scheme where a phase called Key Redistribution replaces the original Path-key Establishment. Suppose that nodes A and C share a common key K1, that B and C share a common key K2, and that A and B have no key in common. In the Key Redistribution phase, A analyzes the lists of key IDs received, determining that K2 could be used to establish a link between B and itself. Then, A asks C to send K2(encrypted with K1) and to delete this key from its memory. If A gains the ownership of K2 in this manner, it now has a common key with B. If C refuses to send the key(e.g., C is already using K2 in one of its direct links, or K2has already been moved), A needs to try other keys and/or nodes until it gets a common key with B or all alternatives are exhausted. In the latter case, A chooses an unused key K3 from its key chain and sends it to node C, which in return computes a reinforced key K2+3 = Hash(K2||K3).
This new key is encrypted with K1 and with K2, and then both encryption results, (EK1(K2+3)) and EK2(K2+3)), are sent back to A. Node A decrypts the EK1(K2+3) and adds K2+3 to its own key chain; meanwhile, EK2(K2+3) is forwarded to node B, which takes the same procedure. At the end of this process, nodes A and B will finally have a common key K2+3. Besides, after the Key Redistribution phase finishes, A has a common key with all its neighbours and, hence, some unused keys can be removed at random in order to reduce memory usage and the information that would be leaked by its capture. According to the simulations presented in [18], the proposed modification leads to a higher key connectivity than the one obtained with the Basic Scheme's original Path-key Establishment. This behaviour is even more accentuated when the keys moved from one node to another are taken into account in several iterations of the Key Redistribution: in this case, these keys not only allow the communication between a pair of nodes, but also create new opportunities for key establishment in the entire neighbourhood. The scheme also improves the resilience of Basic Scheme, since it employs reinforced keys similar to those generated in the Q-Composite Scheme. Nonetheless, this approach still incurs in considerably high communication overheads, especially when the key chain updates are constantly informed to the neighbourhood in order to allow further Key Redistribution iterations.

4.8. Establishing pair wise keys

The Pair wise Key Establishment protocol [27] is a solution that avoids some of the communication overhead involved in the Shared-Key Discovery phase. For this, a unique ID is attributed to each of the n nodes in the network, and each of the |P| keys in the key pool receives an ID between 0 and |P|- 1. The IDs of the keys that are assigned to a node are then chosen by a Pseudo-Random Function (PRF) which, using the node's ID as seed, outputs a total of k integers between 0 and |P|- 1. Hence, in the Shared-Key Discovery phase, any node A can determine which keys another node B possesses simply by applying the same PRF on the ID of B. When compared with the Basic Scheme, where each node would have to broadcast a total of k key IDs (i.e., its entire key chain), this approach is significantly more communication-efficient. Moreover, like in the Multipath Key Reinforcement, this scheme also addresses the issue of many direct links being protected by a same key Ki. The main difference between the two solutions is that, in order to deliver the key updates, the Pair wise Key Establishment protocol uses logical paths, i.e., paths composed both by direct and indirect links not necessarily physically disjoint. Despite this difference, the resulting security improvement and communication overhead introduced by both strategies are similar. Thus, when compared to the combination of Basic Scheme and Multipath Key Reinforcement, the contribution of the Pair wise Key Establishment is mainly a trade-off between processing and communication efficiency (which is desirable in sensor networks) during the Key-Distribution phase. The Cooperative Pair wise Key Establishment protocol [24] builds on the Pair wise Key Establishment scheme in order to provide a more resilient solution at the cost of additional processing and communication overheads.

4.9. Addressing multiple deployments

One vulnerary ability of many probabilistic schemes, such as the Basic Scheme, is that the continuous usage of a same key pool for different generations of nodes facilitates the task of compromising the network's communications. This happens because the keys captured at any time can be used during the whole network's lifetime. Some proposals for addressing this issue, which is also considered in the LKMS scheme [12], are discussed in the following. In the Robust Key (RoK) pre-distribution scheme [6], the key chains of each generation i are constructed from two different key pools, Pif (the ‘‘forward key pool") and Pib (the ‘‘backward key pool"). Such pools are built with random keys, and updated for each generation.

5. Polynomial-based schemes

Assume a network adopting l-bit keys in the finite field GF(q), where q is a sufficiently large prime number. The Polynomial Based Key Pre-distribution [3], also known as Scheme, uses a randomly generated k-degree polynomial f

(x; y)= ∑λij=0aijxiyj over GF(q) satisfying the property f(x,y) = f(y,x). During the pre-distribution phase, each sensor i receives a polynomial share f(g), i.e., a partially evaluated polynomial corresponding to its index i. In this manner, the space occupied by the polynomial loaded into each node is
(λ + 1) log2(q). With this information, node I can establish a common key with node j by evaluating (i,y) at node j and vice versa: the key generated assumes the form
Ki,j = f(i, j) = f(j,i).

This solution shares some interesting features with Blom's Scheme [2], such as the k-secure property, the perfect key-connectivity and the ability to identify and authenticate individual nodes. However, since this schemes non-interactive, it does not add communication overhead to the key establishment process. Thus, the main constraints in this solution are the memory required for storing polynomial shares and the processing power needed for its operations (exponentiations and multiplications).

Some of these limitations are addressed by more recent schemes, discussed in the following.

5.1. Polynomial pool-based key pre-distribution

Liu and Ning [19] have proposed the Polynomial Pool based Key Pre-distribution as a combination of the key-pool paradigm with the Blonde's Scheme [3]. In fact, this solutions analogous to the Multiple-Space Key Pre-Distribution Scheme [11] when polynomials are used instead of matrices. Generally speaking, this scheme uses a set containing randomly generated k-degree polynomials of the form f (x, y)= ∑λij=0aijxiyj over GF(q), for a sufficiently large prime q. The polynomial shares distributed to the network's nodes are taken from this set. Liu and Ning proposed two instances of schemes employing such polynomials.

In the first instance, Random Subset Assignment [19],each node receives a subset of τ (2 ≤ s <w) polynomials, which could be selected in two different ways. In the pre distribution approach, each node would be loaded with the IDs of all other nodes with which they share a common polynomial. This strategy could simplify the Shared-Key Discovery phase, but would impair the addition of new sensors into the network after the initial deployment. For this reason, the real-time discovery approach is preferred. In this case, each polynomial receives a unique ID. As a result, during the Shared-Key Discovery phase, nodes can find their common polynomials by broadcasting their ID lists, or else puzzles solvable only through the knowledge of these polynomials (improving security). Afterward, in the Path-key Establishment phase, neighbouring nodes i and j that are unable to establish a direct link can compute a shared key in the following manner. Node i broadcasts are quest message, containing both i's and j's lists of polynomial IDs. Any node that receives this request and is able to establish a key Ki with i and a key Kj with j replies with a message containing two copies of a randomly generated key Ki,j, one encrypted with Ki and the other with Kj. Nodes i and j recover this new pair wise key from the received message.

The second instance, called Grid-Based Pre-distribution[19], involves the construction of an m x m 2-dimensionalgrid from a set of 2m polynomials {fcα(x, y), frβ(x, y)}, where 1≤β, β≤m, m = ┌√n┐ and n is the size of the network. Each row α in the grid is associated with the polynomial fra(x,y), and each column β is associated with the polynomial fcβ(x,y). Each sensor i in the network is assigned to a unique intersection (α,β) in this grid, which determines the node's ID (IDi = <ci,ri> = <α,β>) and the polynomial shares it receives (fcα(x, y), frβ(x, y)).

To facilitate the Path-key Establishment phase, the nodes are densely placed in a rectangular area of the grid. During the Shared-key Discovery phase, nodes i and j have a polynomial share in common if ci = cj or if ri = rj: this share is fci(x, y) or fri(x, y) respectively. If ci != cj and ri != rj, nodes i and j need to perform the Path-key Establishment protocol, which consists in finding a number of non-compromised nodes in the network whose coordinates in the grid allow the construction of a path between i and j. The original Grid-Based scheme can be further extended yielding the Hypercube-Based Scheme [22], which adopts aτ-dimensional grid instead of a 2-dimensional one. In general, the higher the value of τ, the lower the key-connectivity achieved during the Shared-key Discovery phase, the higher the memory overhead introduced, but the higher the security against node capture. Indeed, this generalized scheme assures that any pair of nodes can establish a common key using the Path-key Establishment protocol despite the number of compromised nodes in the network, as long as the adequate λ and m = ┌τ√n┐ parameters are chosen [22, Section 5.5.2].Some features of these solutions are inherited from Blundo's Scheme [3], such as the λ-security property, node-to-node authentication capabilities and the possibility of dynamically add nodes into the network. There are, though, relevant differences. The amount of data stored and exchanged by the nodes is increased in the Random Subset Assignment, but the resulting resilience against the capture of random nodes in the network is improved. Besides, when compared to this latter solution, both the Grid-Based Pre-distribution and the Hypercube-Based Scheme present additional advantages: due to the intrinsic link between the ID of a node and the polynomial shares it carries, there is no need to broadcast the IDs of these polynomials, which results in smaller communication overheads; moreover, a superior key-connectivity is achieved when no nodes are compromised. Nonetheless, all three approaches also share limitations with Blonde’s Scheme, such as the usage of complex operations.

5.2. PIKE

The Peer Intermediaries for Key Establishment (PIKE) [7]approach shares some similarities with the Grid-Based scheme [19]. For a network of size n, each node receives unique ID (x,y) corresponding to the coordinates of a ┌√n┐ x ┌√n┐ matrix. Each node (x,y) then receives a pair wise key with every node in the same row or column, totalizing2(┌√n┐ - 1) pair wise keys. After deployment, any pair of nodes (xA,yA) and (xB,yB) that do not have a pair wise keys in common can use node (xA,yB) or (xB,yA) as intermediate to establish an indirect link. Since PIKE adopts only pair wise keys, it displays a high security level. Moreover, PIKE’s memory overhead is O(┌√n┐) instead of the O(n) overhead observed in solutions such as [22,27] for a fixed security level. However, since there are at most two nodes that can act as intermediaries for each indirect link formation, this solution often involve network-wide communications during its Path-key Establishment phase. As a consequence, PIKE introduces communication costs of O(┌√n┐), which seriously impairs its applicability in large-scale scenarios.

6. Improving efficiency by using deployment knowledge

The efficiency of the previous schemes can be improved in scenarios where the final position of the nodes can be determined at some extent and their mobility is reduced or non-existent. As this is not the most general scenario, the usefulness of these proposals is restricted to a certain number of applications where this condition is satisfied.

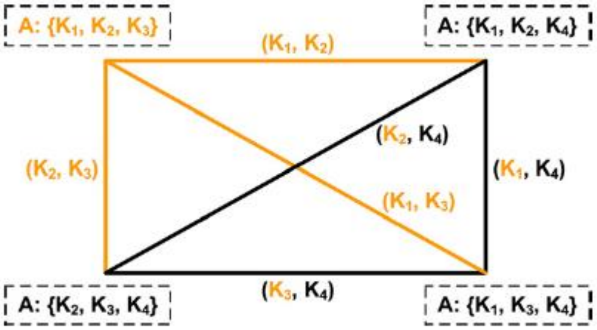
However, whenever the position of nodes (or groups of them) can be estimated with enough precision, this trade-off between flexibility and efficiency may be very advantageous.

6.1. Extending probabilistic schemes

Du et al. [10] combine the Basic Scheme [13] with deployment knowledge in the Group-Based Deployment Scheme. They assume that the nodes are deployed in-groups of c sensors over a u _ v rectangular area, such that the deployment points of the u x v groups

Gi,j (1≤i ≤ u,1 ≤j ≤v) form a rectangular grid. During the Key Pre-distribution phase, the original key pool P is divided in smaller pools Pi; j, each of which is associated to a different group Gi,j in such a manner that any pair of pools used by nearby groups have a big overlap (i.e., many keys is common),while the pools for distant groups have a small or no overlap. After the keys and their IDs are loaded into the sensors from the appropriate pools, the Shared-key Discovery and Path-key Establishment phases proceed as in the Basic Scheme.

For a desired connectivity level and network size, the Group-Based Deployment Scheme associates each node to a pool that is smaller than the one needed by the Basic Scheme alone (e.g., 1/(u.v)). As a result, when compared to the latter, this proposal is more efficient in terms of memory usage and bandwidth occupation, since each node stores and broadcasts less keys; additionally, the number of keys recovered thorough the capture of a single node is reduced and these keys can be used basically to compromise the communication of neighbouring groups (not the entire network), leading to a better resilience. Hence, it trades flexibility for security and efficiency.



Example of key graph for the Complete Graph Design [28] where P ={K1, K2, K3, K4). Highlighted keys and links are the ones compromised by the capture of node A

6.2. Extending polynomial-based schemes

Liu and Ning [20, 21] also have proposed a solution combining the Closest Pair wise Keys Scheme and Blundo’s Scheme [3]. The resulting Polynomial Closest Pair wise scheme involves the partition of the deployment region in g small areas, called cells, each of which is linked to abbreviate λ-degree polynomial. Thus, if a node A is going to be deployed in cell CA, that node is loaded with a set of polynomial shares associated to the cells that are closest to CA, instead of receiving pair wise keys. During Shared key Discovery, each node A broadcasts its ID and the IDs of the polynomials it carries, allowing any neighbour node B that shares a polynomial with A to compute the shared key KA,B as in Blonde’s Scheme. The Triangle-Based scheme proposed in [9] can be seen as an instance of this approaching which the cells are triangles.

A similar solution that combines Blundo’s Scheme [3] with deployment knowledge is the Hexagonal Group-based Key Management (HGKM) [4]. This scheme assumes that the nodes are deployed in groups, following a Gaussian distribution around the point of deployment. The network

is then modelled as a hexagonal grid that follows this distribution pattern and covers all sensor nodes. Every hexagon receives a coordinate (i, j) and is associated to three cells
C1 = {(i, j), (i + 1,j), (i + 1,j + 1)},
C2 = {(i, j), (i, j - 1), (i - 1, j)}and
C3 = {(i, j), (i - 1, j + 1), (i, j + 1)}.
Finally, each cell receives a k-degree polynomial and, thus, each sensor stores three polynomial shares for establishing keys in the same manner as in Blonde's Scheme. More recently, a small modification to HGKM's key establishment process has been proposed [5]. In the resulting scheme, which we call Nonce-based-HGKM (NHGKM),every node A receives a nonce NA together with its polynomial shares. These noises are broadcast during the scheme's Shared-Key Discovery phase, and two nodes A and B having shares of the same polynomial f can then compute a common key KA,B = f(IDA - NA,IDB - NB). As a result, even if adversaries are able to capture a set S (with |S|> k) of nodes having shares of the same polynomial, they would still need to know the noises NA and NB in order to compromise the communication between nodes A and B that are not in S. This added protection provided by the noises is limited, though, since they are broadcast as plaintext during Shared-Key Discovery. The above solutions can be seen as extensions of the Polynomial Pool-based Key Pre-distribution [19] where the assignment of polynomials from the pool is not random, but rather based on the nodes' expected locations for achieving a higher key connectivity. The communication overhead is reduced, staying slightly below other location-based solutions such as [20, 21]. The security achieved depends on the number of nodes per cell. Larger cells result in a larger number of sensors sharing the same polynomial. This results in better key connectivity, but also lead to lower resilience, since each cell inherits the k-security property from Blonde's Scheme and the capture of one node reveals information about three cells.

Future Scope

WSNs, such as sparse WSNs (Anastasia et al., 2009) and mobile WSNs (Chuang et al., 2007) are still open research fields. Moreover, the authentication delay introduced in key-chain-based broadcast authentication mechanisms cannot satisfy real-time applications. Furthermore, there are many potential ways to disrupt the time-synchronization required in broadcast authentication techniques. Hence, the development of time-synchronization independent broadcast authentication mechanisms is another promising area for researchers.

References

[1]. R. Anderson, H. Chan, A. Perrig, Key infection: smart trust for smart dust, in: Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP'04), IEEE Computer Society, Washington, DC, USA, 2004, pp. 206–215.

[2]. R. Blom, An optimal class of symmetric key generation systems, in: Proceedings of the EUROCRYPT 84 Workshop on Advances in Cryptology: Theory and Application of Cryptographic Techniques, Springer, New York, NY, USA, 1985, pp. 335–338.

[3]. C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung, Perfectly secure key distribution for dynamic conferences, in: LNCS, vol. 740, Springer, New York, NY, USA, 1993, pp. 471–486.

[4]. N. Canh, Y.-K. Lee, S. Lee, HGKM: a group-based key management scheme for sensor networks using deployment knowledge, in: Proceedings of the Sixth Annual Communication Networks and Services Research Conference (CNSR'08), IEEE Computer Society, Los Alamitos, CA, USA, 2008, pp. 544–551.

[5]. N. Canh, P. Truc, T. Hai, L. Hung, Y. Lee, S. Lee, Enhanced group-based key management scheme for wireless sensor networks using deployment knowledge, in: Proceedings of the Sixth IEEE Consumer Communications and Networking Conference (CCNC'09),IEEE Computer Society, Los Alamitos, CA, USA, 2009, pp. 1–5.

[6]. C. Castelluccia, A. Spognardi, RoK: a robust key pre-distribution protocol for multi-phase wireless sensor networks, in: Proceedings of the Third International Conference on Security and Privacy in Communications Networks (SecureComm'07), IEEE Computer Society, Los Alamitos, CA, USA, 2007, pp. 351–360.

[7]. H. Chan, A. Perrig, and PIKE: peer intermediaries for key establishment in sensor networks, in: Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05), vol. 1, IEEE Communications Society, Washington, DC, USA, 2005, pp. 524–535.

[8]. H. Chan, A. Perrig, D. Song, Random key pre-distribution schemes for sensor networks, in: Proceedings of the 2003 IEEE Symposium on Security and Privacy (SP'03), IEEE Computer Society, Washington, DC, USA, 2003, pp. 197–213.

[9]. H. Dai, H. Xu, Triangle-based key management scheme for wireless sensor networks, Frontiers of Electrical and Electronic Engineering in China 4 (3) (2009) 300–306.

[10]. W. Du, J. Deng, Y. Han, S. Chen, P. Varshney, A key management scheme for wireless sensor networks

using deployment knowledge. Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04), vol. 1,IEEE Computer Society, Los Alamitos, CA, USA, 2004, pp. 586–597.

[11]. W. Du, J. Deng, Y. Han, P. Varshney, J. Katz, A. Khalili, A pairwise key pre-distribution scheme for wireless sensor networks, ACM Transactions on Information and System Security 8 (2) (2005)228–258.

[12]. B. Dutertre, S. Cheung, J. Levy, Lightweight key management in wireless sensor networks by leveraging initial trust. Technical Report SRI-SDL-04-02, System Design Laboratory, SRI International, April 2004.

[13]. L. Eschenauer, V. Gligor, A key-management scheme for distributed sensor networks, in: Proceedings of the Ninth ACM Conference on Computer and Communications Security (CCS'02), ACM, New York, NY, USA, 2002, pp. 41–47.

[14]. HART. HART7 specification. September 2007. Available from:<www.hartcomm.org>.

[15]. S. Hussain, M. Rahman, L. Yang, Key pre-distribution scheme using keyed-hash chain and multipath key reinforcement for wireless sensor networks, IEEE Computer Society, Los Alamitos, CA, USA,2009, pp. 1–6.

[16]. D. Hwang, B. Lai, I. Verbauwhede, Energy-memory-security tradeoffs in distributed sensor networks, in: ADHOC-NOW, Springer, Berlin/Heidelberg, 2004, pp. 7081.

[17]. B. Lai, S. Kim, I. Verbauwhede, Scalable session key construction protocol for wireless sensor networks, in: IEEE Workshop on Large Scale Real-Time and Embedded Systems (LARTES), IEEE Computer Society, Washington, DC, USA, 2002.

[18]. C.-F. Law, K.-S. Hung, Y.-K. Kwok, A novel key redistribution scheme for wireless sensor networks, in: IEEE International Conference on Communications (ICC'07), IEEE Computer Society, Washington, DC,USA, 2007, pp. 3437–3442.

[19]. D. Liu, P. Ning, Establishing pairwise keys in distributed sensor networks, in: Proceedings of the 10th ACM Conference on Computer and communications Security (CCS'03), ACM, New York, NY, USA,2003, pp. 52–61.

[20]. D. Liu, P. Ning, Location-based pairwise key establishments for static sensor networks, in: Proceedings of the First ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03), ACM, New York, NY, USA, 2003, pp. 72–82.

[21]. D. Liu, P. Ning, Improving key pre-distribution with deployment knowledge in static sensor networks, ACM Transactions on Sensors and Networks 1 (2) (2005) 204–239

[22]. D. Liu, P. Ning, R. Li, Establishing pairwise keys in distributed sensor networks, ACM Transactions on Information and System Security 8(1) (2005) 41–77.

[23]. R. Merkle, Secure communications over insecure channels, Communications on ACM 21 (4) (1978) 294–299.

[24]. R. Di Pietro, L. Mancini, A. Mei, Random key-assignment for secure wireless sensor networks, in: Proceedings of the First ACM workshop on Security of Ad Hoc and Sensor Networks (SASN'03), ACM, New York, NY, USA, 2003, pp. 62–71.

[25]. T. Shan, C. Liu, Enhancing the key pre-distribution scheme on wireless sensor networks, in: IEEE Asia-Pacific Conference on Services Computing, IEEE Computer Society, Los Alamitos, CA, USA,2008, pp. 1127–1131.

[26]. J. Spencer, The Strange Logic of Random Graphs Series: Algorithms and Combinatorics, vol. 22, Springer, Berlin/Heidelberg, 2001.

[27]. S. Zhu, S. Xu, S. Setia, S. Jajodia, Establishing pairwise keys for secure communication in ad hoc networks: a probabilistic approach, in: Proceedings of the 11th IEEE International Conference on Network Protocols (ICNP'03), IEEE Computer Society, Washington, DC, USA,2003, pp. 326–335.

[28]. A. Gupta, J. Kuri, Deterministic schemes for key distribution in wireless sensor networks, in: Proceedings of the Third International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08), IEEE Computer Society, Washington, DC, USA, 2008, pp. 452–459.

[29]. Teena Suneja, Parveen Yadav, Cluster Head Scheme for Wireless Sensor Networks and Security Challenges, International Journal of Computational Engineering and Management IJCEM,vol. 15, Issue 5, Sep. 2012.

Ms. Urvashi Sangwan completed her M.Tech.(CSE) from Ch. Devi Lal University, Sirsa. Now she is Assistant Professor in Vaish College of Engineering, Rohtak (Haryana) and having about eight years experience in field of computer science studies. The field of interest of author is informatics and network studies and digital processing.

.