A Key Distribution Scheme for Mobile Wireless Sensor Networks: Q-s-Composite

# A Key Distribution Scheme for Mobile Wireless Sensor Networks: $q$-$s$-composite

Filippo Gandino, *Member, IEEE,* Renato Ferrero, *Member, IEEE,*
and Maurizio Rebaudengo, *Senior Member, IEEE*

**Abstract**—The majority of security systems for wireless sensor networks are based on symmetric encryption. The main open issue for these approaches concerns the establishment of symmetric keys. A promising key distribution technique is the random predistribution of secret keys. Despite its effectiveness, this approach presents considerable memory overheads, in contrast with the limited resources of wireless sensor networks. In this paper, an in-depth analytical study is conducted on the state-of-the-art key distribution schemes based on random predistribution. A new protocol, called *$q$-$s$-composite*, is proposed in order to exploit the best features of random predistribution and to improve it with lower requirements. The main novelties of $q$-$s$-composite are represented by the organization of the secret material that allows a storing reduction, by the proposed technique for pairwise key generation, and by the limited number of predistributed keys used in the generation of a pairwise key. A comparative analysis demonstrates that the proposed approach provides a higher level of security than the state-of-the-art schemes.

**Index Terms**—Key management, WSN, random predistribution

## 1 Introduction

Wireless sensor networks (WSNs) are becoming a very popular pervasive technology. They normally consist of many low-cost and low-power devices that are able to sense the environment, process information, and transmit messages by means of wireless communication. WSNs are used for different applications, such as indoor tracking [1], labor risk prevention [2], infrastructure monitoring [3] and health-care [4]. However, because of their limited resources, WSNs are exposed to many security threads, such as eavesdropping, hardware tampering and false messages injection [5]. Therefore, effective security systems that are compliant with the specific characteristics of a WSN are required.

*Symmetric cryptography* is commonly used in order to provide security in WSNs. It requires that two neighboring nodes (i.e., in the reciprocal communication range) share a common *key* for the encryption and decryption of the exchanged messages and/or a key for their authentication. The establishment of the symmetric keys is called *key distribution* [6]. Although there are systems to detect compromised nodes and recover security [7], [8], [9], limiting the effects of possible attacks remains a fundamental goal.

In the literature, there are many key distribution schemes for WSNs. Among them, two basic approaches can be identified: the *Plain global key scheme* (PGK) and the *Full pairwise keys scheme* (FPWK) [6]. In both approaches all keys are predistributed before the deployment (e.g., the system administrator stores this information in the memory of the nodes). Therefore, these approaches do not involve communication or computational overheads. Moreover, additional information, such as deployment knowledge, is not required. In PGK, a unique key is used by all nodes, while in FPWK, each node shares a specific key with each other node in the network, so any possible link has its own key. PGK requires a limited memory overhead, but its level of security is low, in fact an opponent with a stolen key can eavesdrop on all links and to introduce new nodes in the network. The level of security with FPWK is higher, since an opponent with a stolen key can only eavesdrop on one link. However, each node stores a key for each other node and a large memory area is used. Therefore, FPWK can only be adopted for small networks while PGK provides a low level of security independently of the size of the network.

Much research in recent years has focused on key distribution, and several new schemes have been proposed. Many approaches have strong requirements and can be applied only to some kinds of networks, while other schemes are designed for WSNs with specific characteristics (e.g., heterogeneity [10], [11], deployment knowledge [12], [13]). However, there are approaches that can be applied to generic WSNs. *Random key predistribution* [14], [15] represents an effective technique based on the preliminary distribution of secret material to the nodes. Every node receives a *ring* of keys that are randomly selected from a general *pool* of keys. If two nodes have one or more keys in common, they can use them for cryptographic operations. An important key distribution scheme based on random key predistribution is the *$q$-composite* scheme [15]. Its main characteristic is that any couple of nodes can establish a symmetric key only if they share at least $q$ *starting keys*. The two nodes execute a hash function on the concatenation of all shared starting keys and use the result as a new *pairwise key*.

In the current paper, an analysis based on the statistical distribution of the keys identifies the best configuration of $q$-composite, that can provide the best performance with


*F. Gandino, R. Ferrero and M. Rebaudengo are with the Dipartimento di Automatica e Informatica, Politecnico di Torino*
*E-mail: see http://www.polito.it/search/index.php?lang=en*

respect to state-of-the-art key distribution schemes without special assumption (e.g., static or small networks). Although $q$-composite provides a high level of security, it is characterized by a large memory requirement: storage represents a critical point for key management schemes based on probabilistic predistribution, since with a larger available storage it is possible to distribute a larger quantity of keys and to improve the resilience of the network. Similarly, with a fixed available memory, resilience can be increased by reducing the memory requirements of the key management scheme. In order to address this issue, a new protocol called *$q$-$s$-composite* is proposed here. The main novelty of the proposed scheme is that the quantity of starting keys used for the generation of a pairwise key is limited by an upper bound (called $s$). Moreover, instead of generating a pairwise key per neighboring node, each node stores the information for the key generation and computes the pairwise key when it is required by the security system of the network (e.g., to encrypt the messages). In order to avoid an additional computation overhead, a light key generation technique based on the bit-wise XOR operation is introduced. Although the proposed approach uses fewer keys per link, thus potentially limiting the security, an in-depth analysis shows that the capability of storing a larger quantity of keys per node not only compensates but even drastically overcomes this apparent drawback, by improving the resilience of the nodes against adversaries with compromised secret material.

The organization of the rest of the paper is the following: in Sect. 2 related works are described. Sect. 3 presents the proposed scheme. In Sect. 4 the proposed approach is evaluated, while it is compared with state-of-the-art schemes in Sect. 5. Finally, conclusions are drawn in Sect. 6.

## 2 Related Work

In this section a brief description of the main state-of-the-art approaches is presented. For a more in-depth description it is possible to read existing surveys on key distribution in WSNs [6], [16].

### 2.1 Random key predistribution approaches

Random key predistribution basically consists in the generation of a large quantity of secret material and in the random distribution of a part of this material to each node. The first random key predistribution approach, hereinafter called *EG*, has been proposed by Eschenauer and Gligor [14]. In EG, a pool containing $p$ keys is generated before deployment. Then, a ring containing $r$ keys that are randomly selected from the pool is picked up by each node. During the network initialization, each node checks if there are shared keys with other neighboring nodes. The values of $p$ and $r$ determine the probability of establishing a link between two nodes and the quantity of secret keys that an adversary can obtain by compromising a node.

A subsequent scheme based on random key predistribution is the *$q$-composite random key predistribution* [15], hereinafter called QC or 1C for $q = 1$, 2C for $q = 2$, etc. With this scheme two nodes have to share at least *$q$ starting keys* to establish a link. They generate a pairwise key by performing a hash function on the concatenation of all shared starting keys. According to the analysis provided in [15], if a small quantity of nodes is compromised, QC provides a higher level of security than EG, especially if the value of $q$ is large.

The main drawback of QC with respect to EG is the larger quantity of memory required. In EG, the starting keys can be directly used as pairwise keys; in contrast, in QC new pairwise keys must be generated and stored. Therefore, by using the same parameters $p$ and $r$, QC provides a higher level of security, but it requires more memory than EG. However, by using the same quantity of memory and by guaranteeing the same connectivity, QC may use values of $p$ and $r$ that cannot guarantee the same security level.

The *Unital-based key predistributed scheme* (UKP) [17] is configured according to two parameters: a prime power $m$ and $t$. UKP allows to extract blocks of elements with specific characteristics:

- a pool is composed of $p = m^3 + 1$ keys,
- there are $m^2\left(m^2 - m + 1\right)$ blocks of keys,
- each block is composed by $m + 1$ keys,
- the same key is present in $m^2$ blocks,
- the same set composed by more than one key cannot be present in more than one block,
- each node receives $t$ disjointed blocks, so $r = t(m+1)$ keys,
- the maximum quantity of nodes is $\frac{m^2}{t}\left(m^2 - m + 1\right)$.

Moreover, the authors calculate that $t \sim \sqrt{m}$ provides the best performance. This approach provides a good level of resilience and is highly scalable, but its level of connectivity is low, so it requires a high value of $t$ and a large memory in order to compensate the connectivity.

In [18], a scheme is proposed with a key establishment mechanism similar to 1C. However, the life of the network is split in phases. After each phase, all nodes apply a hash function on their keys in order to generate a new ring. Therefore, an adversary that compromises a node cannot achieve information about the pairwise keys computed in the previous phases. This scheme is compliant with mobile nodes and has a higher computational overhead with respect to 1C.

### 2.2 Global Master Key

In the class of schemes based on a global master key, all nodes share a master key that is used to establish the final pairwise keys.

The main scheme based on a global master key is the *Symmetric-key key establishment* (SKKE), which is adopted by ZigBee[1]. In this scheme, a node A starts the key establishment by sending a random number ($C_A$). In order to generate a common secret, a node B, after receiving the initial message, computes a new random number ($C_B$) and executes a keyed hash function with the global master key on the concatenation of the subsequent data: $ID_B$, $ID_A$, $C_B$ and $C_A$. After generating the common secret, node B executes a hash function on this secret in order to generate the pairwise key. Then, node B sends back its identifier $ID_B$, the random number $C_B$ and the *Message authentication code* (MAC), calculated on the concatenation of the subsequent

1. ZigBee Specification 1.0, June 2005, ZigBee Alliance

data: a constant number $k_1$, $\mathrm{ID}_B$, $\mathrm{ID}_A$, $\mathrm{C}_B$ and $\mathrm{C}_A$. Then, node A calculates the pairwise key, checks the MAC and sends to node B an MAC generated from the same data concatenated to a second constant number $k_2$.

### 2.3 Transitory Master Key

In the class of schemes based on a transitory master key, after the deployment, each node shares the transitory master key, which is used to establish the final pairwise keys. However, in contrast to the global master key, each node deletes the master key after a time-out. This class of schemes is based on the assumption that an adversary cannot compromise a node in less than a lower bound of time.

The periods before and after the deletion of the master key are called *initialization phase* and *working phase*. According to the specific scheme, the nodes may be able (or not) to establish new keys within the working phase. This ability is required in order to allow new nodes to be added into the network.

The time-out represents a trade-off between efficiency and security, since if it is short, the nodes will not be able to establish pairwise keys with all their neighboring nodes, but if it is long, the probability that an adversary can steal the master key is higher. If the master key is compromised, an adversary can decode all messages potentially finding all pairwise keys.

LEAP+ [19] is a well-known scheme based on a transitory master key. It can be applied only to static wireless networks. LEAP+ provides four kinds of keys, but the main scheme is based on the establishment of the pairwise keys. All nodes know a keyed pseudo-random function and the master key. Moreover, each node has a private master key, which is generated by executing the pseudo-random function with the master key on the identifier of the node. A pair of nodes within the initialization phase establish the pairwise key by executing the pseudo-random function with the private master key of the first node on the identifier of the second one. A node within the initialization phase can also establish pairwise keys with nodes in the working phase by generating their private master key.

RSDTMK [20] is a scheme that combines Transitory Master Key and Random key distribution. Even RSDTMK can be applied only to static wireless networks. A pool containing $p$ seeds is generated before the deployment. Then, a ring containing $r$ seeds, randomly selected from the pool, is picked up per each node. Moreover, all nodes know a keyed pseudo-random function, a simple permutation function and the master key. Two nodes have to share a seed in order to establish a pairwise key. They select a random number on $\mu$ bits that is used with the permutation function to modify the common seed. Then, they execute the keyed pseudo-random function on the previous result. At the end of the initialization, an additional key is generated per unused key of the ring, in order to be able to establish keys also with nodes later deployed. Therefore, the quantity of keys $\varrho$ stored by a node after the key establishment can be larger than the original ring. Under the assumption that the master key cannot be compromised, this scheme provides a slightly lower level of security than LEAP+. However, if the master key is compromised, RSDTMK still provides a basic level of security.

In [21], another scheme that combines Transitory Master Key and Random key distribution was proposed. It uses two pools: the keys of the first are deleted at the end of the initialization while the keys of the other are used for node adding. The main problem is represented by the memory overhead required to store one ring per pool.

### 2.4 Other Approaches

Given the heterogeneity of WSNs, there is a great variety of approaches with different requirements.

Blom [22] proposed a well-known scheme for key exchange based on matrix multiplication. Each node in the network knows some cryptographic material used to generate a specific pair-wise key for each other node which is able to generate the same key. Two schemes based on Blom's work have been presented in [23] and [24]. In both schemes, a pool of secret matrices $\mathrm{D}_{k\times k}$ is generated off-line. According to Blom's scheme, a matrix of identifiers $\mathrm{G}_{n\times k}$ is generated, and each row is matched to a node. Then, a ring of randomly chosen matrices G is matched to each node. Each secret row of each $(\mathrm{DG})^{\mathrm{T}}$ is given to its corresponding node. After deployment, during network initialization, each node verifies if any neighbors share a matrix with it. The shared-key discovery is performed broadcasting the list of the identifiers of the matrix in the ring.

Many schemes are based on deployment knowledge. Liu and Ning [25] proposed using pre-deployment knowledge to distribute pairwise keys to pairs of nodes that should be close to each other according to their expected location. Moreover, they proposed using post-deployment knowledge if nodes can identify their location, by deleting the keys that are farther from the actual location. Yu and Guan [26] presented a scheme based on Blom's work where the deployment area is divided into hexagons matched to a secret matrix. Each node picks up the rows from the matrices matched to its expected hexagon and to the six adjacent ones. In [27], Younis et al. presented SHELL, a scheme that exploits the position of nodes in order to distribute secret material that is mainly shared with nodes in the same area. In [28], a scheme that uses light deployment knowledge is presented. In this case the deployment knowledge corresponds to an approximated division of the network according to the locations of the nodes.

Some key management schemes are only compliant with heterogeneous WSNs. In [29], Dong and Liu proposed a scheme that requires auxiliary nodes to execute the key establishment. In [10], [30], two schemes are proposed based on nodes with high and low processing capabilities. The nodes with high processing capabilities act as cluster heads. Localized combinatorial keying (LOCK) [31] is another scheme designed for heterogeneous network that uses a base station, cluster heads and regular nodes.

## 3 Proposed Approach

In this section the proposed scheme is presented and its main characteristics are discussed.

### 3.1 Background

The following general considerations on security systems for WSNs should be taken into account: (i) if the security system provides both encryption and signature of messages, two different pairwise keys are required; (ii) if a scheme provides other kinds of keys (e.g., keys for local broadcast), the pairwise key establishment normally represents the core of the scheme, while the establishment of other keys can be considered optional and compliant with the majority of the schemes; (iii) although some schemes use a two-way handshake (i.e., the initiator sends a message to present itself and another node answers by sending an acknowledge) and some others use a three-way handshake (i.e., in addition to the first two messages, the initiator answers to the acknowledge by sending a second acknowledge), the selection of the handshake is normally independent of the characteristics of the scheme.

Therefore, in order to reach a fair analysis, the current description and comparison are focused on the establishment of a pairwise key per link with a two-way handshake.

### 3.2 Notation and Assumptions

The proposed approach is based on the subsequent assumptions:

- there is no deployment knowledge;
- the nodes are homogeneous apart from the sink, with greater resources;
- an adversary can eavesdrop on the whole traffic, inject packets, or replay older messages;
- an adversary can also compromise a node through hardware tampering and obtain all the information it holds.

The list of parameters adopted for the description of the proposed scheme includes:

- the quantity of nodes in the network ($n$);
- the quantity of starting keys in the pool ($p$);
- the quantity of starting keys in each ring ($r$);
- the minimum quantity of shared starting keys required to establish a link ($q$);
- the number of neighboring nodes per node ($v$); in order to simplify the analysis, $v$ is assumed constant.

### 3.3 General description of $q$-$s$-composite (QSC)

In order to solve the limitations of QC due to the large memory requirement, a novel key distribution scheme is here proposed. The new protocol is called *$q$-$s$-composite* (QSC).

In QSC, like in previous schemes based on random key distribution [14], [15], a ring of $r$ starting keys is randomly picked up per each node from a pull of $p$ starting keys. Moreover, like in QC, two neighboring nodes can establish a link only if they share at least $q$ keys.

The first novelty of QSC is given by the parameter $s$, which represents the maximum quantity of starting keys used for a pairwise key establishment. The relationship that associates these parameters is:

$$0 < q \leq s \leq r \leq p \tag{1}$$

According to the values of $q$ and $s$, QSC is called 1-1C for $q = 1, s = 1$, 1-2C for $q = 1, s = 2$, etc.

The second novelty is represented by the mechanism used to generate a pairwise key from $x$ starting keys. A bitwise XOR operation, instead of a hash function, is used on the shared starting keys to compute the pairwise key. For example, if the shared keys are $K_X$ and $K_Y$, the final key is $K_{XY} = K_X \oplus K_Y$.

The third novelty is represented by the *storing organization*. Instead of a pairwise key, the list of the identification numbers of, at most, $s$ starting keys is stored per each link. By using the *Advanced encryption standard* (AES) [32] as a symmetric cryptosystem, the length of the keys should be at least 128 bits. Since the starting keys stored by a node can be identified by their position in the set of keys, if $r < 256$, then 8 bits are enough to identify a starting key. Therefore, if $s < 16$, the required memory area is lower ($s \times 8 < 128$). Since the computational effort required by the XOR operation is negligible, the pairwise keys can be computed immediately before using them. The advantage of the proposed approach is that the memory required per each link is smaller. Therefore, a node can store a larger number of keys ($r$) by occupying the same amount of memory.

### 3.4 Predistribution phase

The parameters of QSC are configured before the network deployment: the rings of $r$ starting keys must be randomly picked up from a pool of $p$ starting keys and loaded into the memory of each node $i$ with its unique identifier ($\mathrm{ID}_i$). Fig. 1 shows an example with 4 nodes. Each node initially stores three starting keys and their identifiers.

### 3.5 Key establishment

The pairwise key establishment among nodes is shown in Fig. 1 with $r = 3$, $q = 2$ and $s = 2$. Every node broadcasts periodically a *hello* message. The node that sends the *hello* is called *initiator*. This message is used to communicate to the neighboring nodes the identifier of the initiator and the identifiers of its starting keys. In Fig. 1, node A broadcasts a message with its identifier $\mathrm{ID}_A$, and the identifiers of its starting keys (in the example: $\mathrm{IDk}_X$, $\mathrm{IDk}_Y$ and $\mathrm{IDk}_W$). When a node, called *receiver*, receives a *hello* message from a new node, then it looks for shared starting keys in the received set of IDs.

If there are less than $q$ shared keys, the receiver stops the handshake, since it is not possible to establish a link between the two nodes. In Fig. 1, node B only shares one key (less than $q$) with A.

If the quantity of shared keys is between $q$ and $s$, the receiver records their identifiers associated with the identifier of the initiator. From this moment a pairwise key corresponding to the bitwise XOR of the shared starting keys is associated with that node. However, the pairwise key is not stored, and it will be generated each time before being used. Then, the receiver replies to the initiator with an *acknowledge* message. This message contains the identification of the receiver and the ID of the shared keys. Moreover, the message is authenticated by a MAC executed with the corresponding pairwise key. Finally, the initiator calculates the pairwise key and checks the MAC of the message. If it is

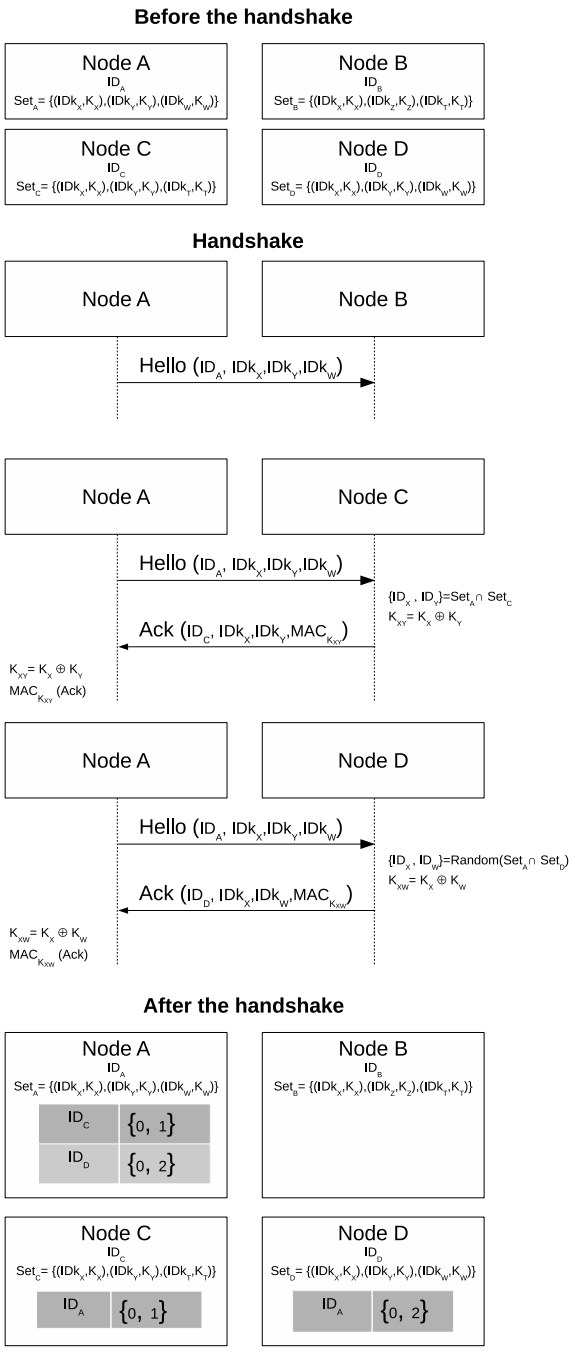


Fig. 1. Pairwise key establishment with $r = 3$, $q = 2$ and $s = 2$.

correct, the initiator stores the identifiers of the used starting keys with the identifier of the receiver. This information will allow the node to compute the pairwise key when it is required. In Fig. 1, node C shares $K_X$ and $K_Y$ (a quantity of keys between $q$ and $s$) with A.

If there are more than $s$ shared keys, the receiver randomly selects $s$ keys. Then, it executes the same routine as the quantity of shared keys would be between $q$ and $s$, but only considering the selected keys. In Fig. 1, node A and D share $K_X$, $K_Y$ and $K_W$ (more than $s$ keys). Node D randomly selects $K_X$ and $K_W$.

### 3.6 Storing organization

After the handshake, each node still stores the initial information. Moreover, per each established link, a node stores the identifier of the neighboring node and the identifiers of the starting keys used for the generation of the pairwise key. In order to save memory, since $r \leq p$, the nodes may only store the positional number, lower than $r$, of the starting keys inside their set. In the example in Fig. 1, there are 2 established links: between A and C, and between A and D. Node A stores a table with two rows: in the first row the identifier of node C is associated with the positions 0 and 1, which correspond to the two starting keys ($K_X$ and $K_Y$) shared with node C; in the second row $ID_D$ is associated with the positions 0 and 2 ($K_X$ and $K_W$) shared with node D. Node B does not store additional information, since it has not established links. Node C and D only store one row, with the information about the link with node A.

### 3.7 Key Update

In order to increase the security of the networks (e.g., against replay-attacks), it is useful to periodically update the pairwise keys. Moreover, if the key update mechanism does not allow to recover the previous keys, an adversary that compromises a node would not be able to decrypt the old messages.

Since in QSC the pairwise keys are not stored, the key updating can be executed by using an additional secret key, called the *updating key*, common to all nodes. The computation of the pairwise keys requires an additional bitwise XOR operation with the updating key. The updating key must be updated periodically by all nodes. It is observed that, in contrast to the majority of the other schemes (e.g., EG and RSDTMK), QSC only updates one key, reducing the required effort.

QSC is compliant with the majority of the existing key updating mechanisms. In the following two eligible approaches are described. First, a distributed mechanism consists in periodically updating the key with a specific frequency, by using a one-way function, such as a hash. In this way, all nodes will update the key at the same time. Then, a centralized mechanism is based on a trust center that periodically updates the key. This approach is used by ZigBee: the center broadcasts the new key, encrypted with the previous network key [33].

## 4 Analysis and Evaluation

In this section, the performance provided by QSC are investigated and compared with the state-of-the-art schemes. Formulas for the analysis of key distribution schemes and in particular for the resilience have been already proposed [34], [20]. In the current approach, the previous analysis is extended and some approximations are replaced by exact formulas.

### 4.1 Mobility and possibility of adding new nodes

In QSC, nodes never delete the secret information required to establish new keys. Therefore, they can establish new links with nodes added to the network after the initial deployment and to establish new links with nodes met by changing the initial position.

TABLE 1
Resilience of the static networks, by considering $x$ compromised nodes

| Phase of the attack | Effect | LEAP+ | RSDTMK |
|---|---|---|---|
| Working | link | 0 | $1-(1-\frac{\varrho}{p2^{\mu}})^x$ |
| | check | 0 | $1-\left(1-\frac{v}{n-1}\left(1-\frac{\binom{p-r}{r}}{\binom{p}{r}}\right)+\left(1-\frac{v}{n-1}\right)\left(1-\frac{\binom{p2^{\mu}-\varrho}{\varrho}}{\binom{p2^{\mu}}{\varrho}}\right)\right)^x$ |
| Initialization | link | 1 | $1-(1-\frac{r}{p})^x$ |
| | check | 1 | $1-\left(\frac{\binom{p-r}{r}}{\binom{p}{r}}\right)^x$ |

TABLE 2
Resilience of the mobile networks, by considering $x$ compromised nodes, Part I

| Effect | FPWK | PGK, SKKE | EG | UKP |
|---|---|---|---|---|
| link | 0 | 1 | $1-(1-\frac{r}{p})^x$ | $\sum_{i=1}^{t^2}\frac{\binom{t^2}{i}\left(\frac{(m+1)^2}{m^3+m+1}\right)^i\left(1-\frac{(m+1)^2}{m^3+m+1}\right)^{t^2-i}}{1-\left(1-\frac{(m+1)^2}{m^3+m+1}\right)^{t^2}}\left(1-\frac{\binom{m^3(m-1)}{xt}}{\binom{m^2(m^2-m+1)}{xt}}\right)^i$ |
| check | 0 | 1 | $1-\left(\frac{\binom{p-r}{r}}{\binom{p}{r}}\right)^x$ | $1-\left(1-\frac{(m+1)^2}{m^2-m+1}\right)^{t^2}$ |

TABLE 3
Resilience of the mobile networks, by considering $x$ compromised nodes Part II

| Effect | QC | QSC |
|---|---|---|
| link | $\sum_{i=q}^{r}\frac{\binom{r}{i}\binom{p-r}{r-i}\binom{p}{r}^{-x}\sum_{j=0}^{i}\binom{i}{j}(-1)^j\binom{p-j}{r}^x}{\binom{p}{r}-\sum_{j=0}^{q-1}\binom{r}{j}\binom{p-r}{r-j}}$ | $\sum_{i=q}^{r}\frac{\binom{r}{i}\binom{p-r}{r-i}\binom{p}{r}^{-x}\sum_{j=0}^{\mathrm{Min}(i,s)}\binom{\mathrm{Min}(i,s)}{j}(-1)^j\binom{p-j}{r}^x}{\sum_{k=q}^{r}\binom{r}{k}\binom{p-r}{r-k}}$ |
| check | $1+\frac{\sum_{j=1}^{q}j\binom{r}{q-j}(-1)^j\binom{p-r+q-j}{r}^x}{\binom{p}{r}^x}$ | $1+\frac{\sum_{j=1}^{q}j\binom{r}{q-j}(-1)^j\binom{p-r+q-j}{r}^x}{\binom{p}{r}^x}$ |

### 4.2 Attacker model

An adversary can conduct a replay attack and he/she can eavesdrop on all the traffic, so he/she potentially knows all the data transmitted in the clear. An adversary knows the identifier of all starting keys ($\mathrm{IDk}_i$) and of the pairwise keys in the network ($\mathrm{IDk}_{1-\ldots-x}$). However, he/she does not know the value of the starting keys ($\mathrm{K}_i$) and of the symmetric pairwise keys ($\mathrm{K}_{1-\ldots-x}$).

An adversary can physically attack a node and retrieve secret information stored by that node. Attacks performed both in the initialization and in the working phases need to be considered in order to compare QSC to the state-of-art schemes based on transitory master key. The following two threats are considered:

- $x$ nodes are compromised in the working phase: an opponent knows all the secret material stored by $x$ nodes, apart from the material potentially erased at the conclusion of the initialization;
- $x$ nodes are compromised within the initialization phase: an opponent knows all the secret material stored by $x$ nodes, including the material potentially erased at the conclusion of the initialization.

After a node is compromised, the adversary can try to use the achieved secret material to eavesdrop on links between other nodes and to inject new illegitimate nodes in the network.

### 4.3 Security of the key generation mechanism

Since the XOR operation is easy to invert, a potential drawback is that by knowing a pairwise key and all except one starting keys used to generate it, an adversary can find the last starting key. However, like in QC, if an adversary compromises a node, further to its pairwise keys, he/she also obtains all starting keys that have been used to generate them. Consequently, the use of the XOR operation does not provide any additional information to adversaries.

### 4.4 Resilience

Resilience represents the ability to resist to the presence of compromised secret material. Table 1, 2 and 3 show the probability that an adversary eavesdrops on a link between uncompromised nodes ("link" rows in Table 1, 2 and 3), and the probability that a false node (not a compromised one) passes an authenticity check ("check" rows in Table 1, 2 and 3), according to one of the analyzed threats.

The resilience against eavesdropping provided by QSC can be considered a general case of the resilience provided by EG and QC. If nodes are compromised during the initialization phase, RSDTMK is equivalent to EG and 1-1C. The

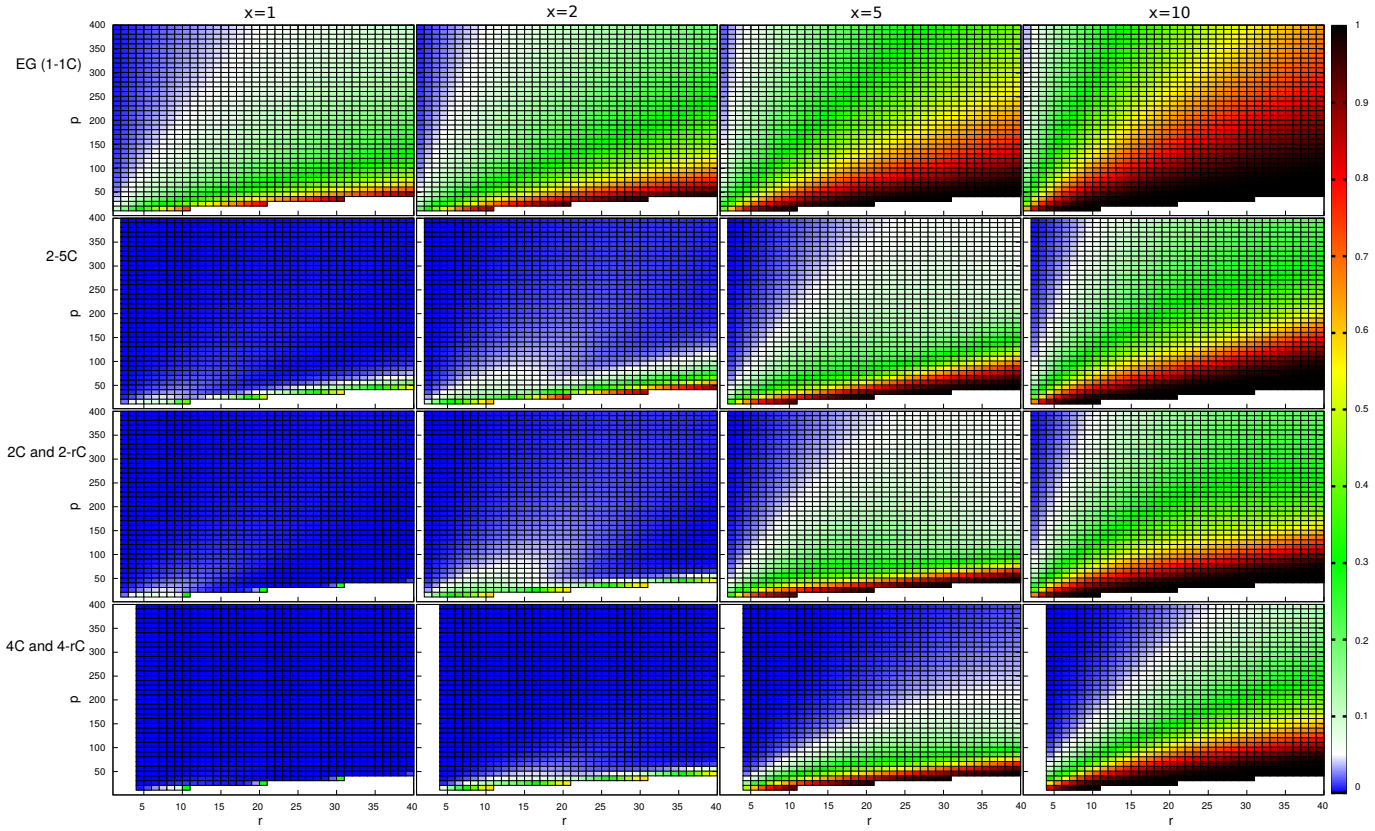


Fig. 2. Probability of eavesdropping with QSC, QC and EG, according to $p$, $r$ and the quantity of compromised nodes $x$.

formula of the probability of eavesdropping on a link for QSC is composed by:

- the probability that the two nodes that communicate on the link to be eavesdropped on share $i$ keys (i.e., their pairwise key is composed by $\text{Min}(i, s)$ starting keys), that corresponds to:
  - the quantity of possible combinations of $i$ shared starting keys among the two rings of the two nodes connected by the link $\left(\binom{r}{i}\binom{p-r}{r-i}\right)$,
  - divided by the total possible valid combinations of shared starting keys between the two rings of the two nodes connected by the link $\left(\sum_{k=q}^{s}\binom{r}{k}\binom{p-r}{r-k}\right)$,
- multiplied by the probability that in the $x$ compromised rings there are all the $\text{Min}(i, s)$ starting keys used by the link, which is calculated as:
  - 1, which corresponds to the first element ($j = 0$) of the summation $\binom{p}{r}^{-x}\sum_{j=0}^{\text{Min}(i,s)}\binom{i}{j}(-1)^j\binom{p-j}{r}^{x}$,
  - minus the probability that at least one of the keys used for the link is not included in the $x$ rings, which corresponds to the subsequent iterations of the summation:
    - at the second step of the summation ($j = 1$) a specific key is not shared (the cases in which more than one key used for the link are not owned by the adversary are considered more than once),
    - at the subsequent steps the redundant cases are corrected alternatively adding and subtracting $\left((-1)^j\right)$ the quantity of their combinations.

If $q = 1$ and $s = 1$, the probability of eavesdropping on a link in QSC is equivalent to the same probability in EG, since each link is protected by a single starting key. If $s = r$, this probability in QSC is equivalent to the same as in QC. Considering to use the same values of $r$ and $p$ for QSC, EG and QC, and the same value of $q$ for QSC and QC, then:

- if $s = r$, QC provides the same level of resilience against eavesdropping like QSC;
- if $s < r$, QC is more resilient against eavesdropping than QSC;
- if $q > 1$, QC and QSC are more resilient against eavesdropping than EG;
- if $q = 1$, QC is more resilient against eavesdropping than EG, and QSC provides the same level of resilience against eavesdropping like EG.

It should be remarked that $s \leq r$, $q \leq r$, $q \leq s$, $r \geq 1$ and that if $r = 1$, then QSC, EG and QC provide the same level of resilience against eavesdropping.

Fig. 2 shows the resilience against eavesdropping provided by EG, QC and QSC. The blue color corresponds to the higher level of resilience. It becomes lighter, up to white, at a probability of eavesdropping equal to 0.05. Increasing the probability of eavesdropping, colors change up to black.

TABLE 4
Connectivity

| FPWK, PGK, SKKE & LEAP+ | 1C, 1-SC, EG & RSDTMK | QC & QSC | UKP |
|---|---|---|---|
| 1 | $1-\frac{\binom{p-r}{r}}{\binom{p}{r}}$ | $1-\frac{\sum_{i=0}^{q-1}\binom{r}{i}\binom{p-r}{r-i}}{\binom{p}{r}}$ | $1-\left(1-\frac{(m+1)^2}{m^3+m+1}\right)^{t^2}$ |

It is possible to observe that with EG the resilience is proportional to $p$ and inversely proportional to $r$. For QSC and QC with $q \geq 1$ the resilience is visibly higher. However, although in general the resilience is higher with a high $p$ and a low $r$, in QC and QSC there exist points of local maximum (e.g., the blue area for $x = 2$ at the right side of the white area). This behavior is due to two opposite phenomena. On the one hand, the probability of eavesdropping on the pairwise keys based on a fixed quantity of starting keys is proportional to $r$ and inversely proportional to $p$. On the other hand, the ratio of pairwise keys based on a large number of starting keys is proportional to $r$ and inversely proportional to $p$. Nevertheless, Fig. 2 shows that the points of local maximum are not so sharp.

Table 2 shows that the best resilience against eavesdropping is provided by FPWK and it corresponds to the theoretical maximum. The worst resilience is provided by PGK and SKKE and it corresponds to the theoretical minimum. The resilience provided by UKP (the formula is approximated and was presented in [17]) is similar to QC, since all shared keys are used to generate the pairwise key. While analyzing the schemes for static networks, it is required to consider separately if $x$ nodes have been compromised in the initialization phase, when they still store the master key, or in the working phase. If $x$ nodes are compromised in the working phase, LEAP+ provides the best theoretical resilience, like FPWK. RSDTMK provides a very high resilience, and according to its parameters, the difference with respect to the maximum could be negligible. If $x$ nodes are compromised in the initialization phase, LEAP+ provides the theoretical worst resilience, like PGK and SKKE, while RSDTMK provides the same resilience as EG.

The formula of the probability of passing the authentication check is the same for QSC and QC, since in both protocols an adversary that shares at least $q$ starting keys with a node can pass its authentication check. The formula for UKP is similar to EG, but it is based on parameters $m$ and $t$. The formula for QSC is calculated as 1 minus the probability that in the rings of the $x$ compromised nodes there are less than $q$ keys shared with the node that will do the authentication check, which is computed as follows:

- each iteration of the summation calculates the quantity of cases in which at most $q-j$ starting keys are shared, computed as:
  - at the first step:
    - the quantity of sets composed by $q-1$ keys inside a ring $\binom{r}{q-j}$,
    - multiplied by the quantity of combinations in which all $x$ rings only include keys out of the ring of the node that will do the check or in a set of $q-1$ keys (at most $q-1$ keys are shared);
  - since this formula counts more than one time the sub-cases in which less than $q-1$ keys are shared, at the subsequent steps the redundant cases are corrected:
    - the correction is alternatively positive or negative $(-1)^j$,
    - and it is equal to the quantity of sets of $q-j$ keys inside a ring $\binom{r}{q-j}$,
    - multiplied by the quantity of combinations in which all $x$ rings only include keys out of the ring of the node that will do the check, or in a set of $q-j$ keys,
    - multiplied by the quantity of cases in which each combination has been redundantly considered $(j)$;
- divided by the total possible combination of starting keys owned by the adversary $\binom{p}{r}^x$,

If $q = 1$, the formula of the probability of passing an authentication check is equivalent in QC and QSC.

Among the schemes for mobile networks, FPWK provides the best resilience against false authentication. QSC and QC provide a resilience better than EG, while PGK and SKKE provide the worst resilience, which is equal to the theoretical minimum. The resilience provided by UKP, according to the distribution mechanism, is similar to the one of QC. LEAP+ provides the best resilience, corresponding to the theoretical maximum, among the schemes for mobile networks if the nodes are compromised in the working phase. RSDTMK provides a resilience that is close to the theoretical maximum and higher than QSC and QC. If $x$ nodes are compromised in the initialization phase, LEAP+ provides the worst resilience, like PGK and SKKE, while RSDTMK provides the same level of resilience like EG.

### 4.5 Connectivity

The connectivity represents the probability to establish a link with a neighboring node. Table 4 shows the formula of connectivity for the state-of-the-art approaches and for the proposed one.

The best connectivity is provided by FPWK, PGK, SKKE and LEAP+, where all possible links are always established. The formula of the connectivity for QSC and QC is the same, since the parameter $s$ does not affect the probability of establishing a key between two nodes. Two nodes can establish a link if they share at least $q$ keys. With the same values for $p$ and $r$, when $q = 1$, QC and QSC provide the same connectivity as EG and RSDTMK. The level of probability corresponds to 1 minus the number

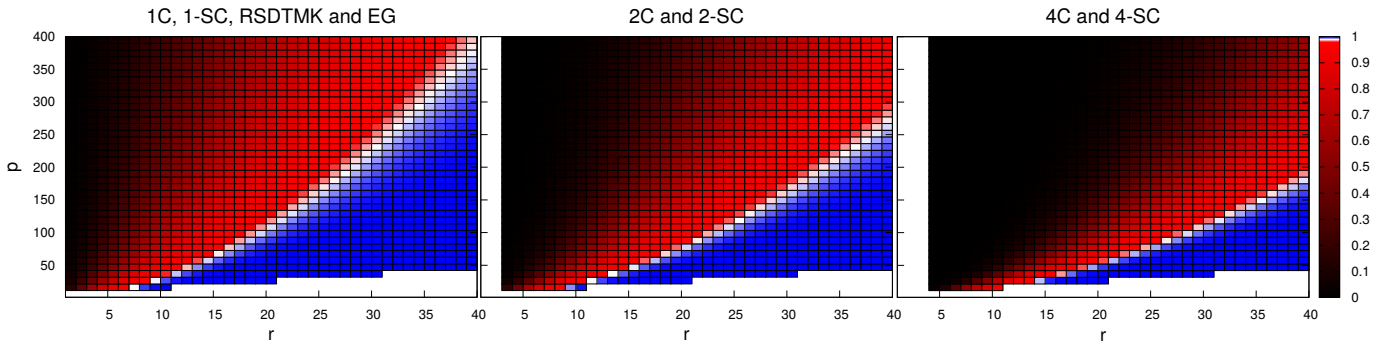


Fig. 3. Connectivity of QSC, QC, EG and RSDTMK according to $p$ and $r$.

TABLE 5
Memory required for secret keys

| Scheme | Prestorage | Working storage |
|---|---|---|
| FPWK | $(n-1)\cdot l_k$ | $(n-1)\cdot l_k$ |
| PGK | $l_k$ | $l_k$ |
| LEAP+ | $2\cdot l_k+$ | $v\cdot(l_k+l_{ID})+l_k$ |
| SKKE | $l_k$ | $v\cdot(l_k+l_{ID})+l_k$ |
| RSDTMK | $r\cdot l_s+l_k+r\cdot l_{sID}$ | $\varrho\cdot(l_k+l_{kID})+v\cdot(l_{ID}+l_{kID})$ |
| EG | $r\cdot(l_k+l_{kID})$ | $r\cdot(l_k+l_{kID})+v\cdot(l_{ID}+l_{kID})$ |
| UKP | $t\cdot(m+1)(l_k+l_{kID})$ | $t\cdot(m+1)(l_k+l_{kID})+v\cdot(l_k+l_{ID})$ |
| QC | $r\cdot(l_k+l_{kID})$ | $r\cdot(l_k+l_{kID})+v\cdot(l_k+l_{ID})$ |
| QSC | $r\cdot(l_k+l_{kID})+l_k$ | $r\cdot(l_k+l_{kID})+v\cdot(s\cdot l_{kID}+l_{ID})+l_k$ |

of combinations of keys in a ring that does not include keys belonging to a second ring $\binom{p-r}{r}$, divided by the total quantity of possible combinations of keys in a ring $\binom{p}{r}$. If $q > 1$, the level of connectivity of QC and QSC is lower. The level of probability corresponds to 1 minus the number of combinations of keys in a ring that includes less than $q$ keys for a second ring $\left(\sum_{i=0}^{q-1}\binom{r}{i}\binom{p-r}{r-i}\right)$, divided by the total quantity of possible combinations of keys in a ring $\binom{p}{r}$. For EG, UKP, QC and QSC the connectivity is equal to the probability that an opponent can pass an authenticity check when one node is compromised. This equality represents a strict constraint, since it is not possible to reach high levels of both connectivity and resilience against false authentication. Fig. 3 shows the connectivity of QSC, QC, EG and RSDTMK according to $p$ and $r$. The blue area in the chart represents a connectivity greater than 0.99, the white area corresponds to 0.99, while for a lower level of connectivity the color moves from red to black (corresponding to 0). It is possible to observe that, with a fixed quantity of keys per ring, a high level of $q$ requires to limit the total quantity of keys in the pool in order to maintain a high level of connectivity. Fig. 4 shows the level of connectivity of UKP according to $t$ and $m$. In order to reach a high level of connectivity, UKP requires low values for $m$ and high values for $t$.

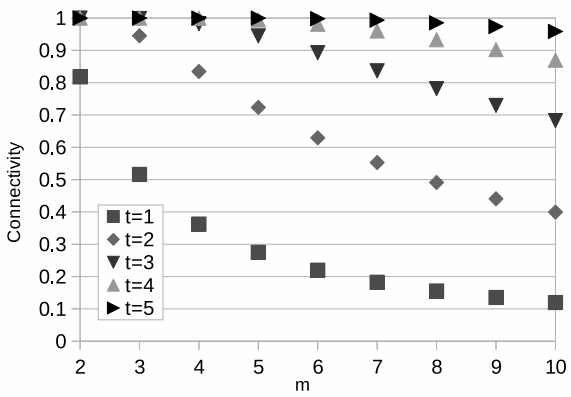


Fig. 4. Connectivity of UKP according to $t$ and $m$.

### 4.6 Storage efficiency

In order to evaluate the storage efficiency of QSC, in comparison with the state-of-the-art schemes, the following parameters are considered:

- length of a key ($l_k$),
- length of a node identifier ($l_{ID}$),
- length of a seed in RSDTMK ($l_s$),
- quantity of pairwise keys stored by each node in RSDTMK ($\varrho$),
- length of a seed identifier in RSDTMK ($l_{sID}$).

Table 5 shows the memory required to store the secret material. The main part of the memory required by EG is used to store $r$ keys. Therefore, the storage constraint limits $r$ and consequently the connectivity. QC needs to store both the starting keys and the final pairwise keys. Moreover, QC stores a one-way function. However, in QC a pairwise key does not need any identifier, since future neighboring nodes will establish new pairwise keys only using the shared starting keys. UKP has a storing organization similar to

TABLE 6
Communication efficiency

| Scheme | Hops | # of messages | Size of transmitted data |
|---|---|---|---|
| FPWK | 0 | 0 | 0 |
| PGK | 0 | 0 | 0 |
| LEAP+ | 1 | 2 | $2 \cdot l_{ID} + l_k$ |
| SKKE | 1 | 2 | $3 \cdot l_{ID} + 4 \cdot l_k$ |
| RSDTMK | 1 | 2 | $r \cdot l_{sID} + l_{kID} + 2 \cdot l_{ID} + l_k$ |
| EG | 1 | 2 | $(r+1) \cdot l_{kID} + 2 \cdot l_{ID} + l_k$ |
| UKP | 1 | 2 | $\left(t(m+1)+t^2\right) \cdot l_{kID} + 2 \cdot l_{ID} + l_k$ |
| QC | 1 | 2 | $2r \cdot l_{kID} + 2 \cdot l_{ID} + l_k$ |
| QSC | 1 | 2 | $(r+s) \cdot l_{kID} + 2 \cdot l_{ID} + l_k$ |

QC. QSC stores the starting keys and the identifiers of the shared keys with the identifier of the corresponding node. Moreover, QSC stores the updating key. LEAP+ stores $v+1$ keys, their node identifiers and a pseudo-random function. RSDTMK stores $\varrho$ keys (MAX$(r, v) \leq \varrho < v + r$), their identifiers, a one-way function and a permutation function.

In order to provide a pessimistic analysis for the proposed approach, the quantity of key identifiers stored by QSC is always considered the maximum ($s$) in the storage analysis, although it can be lower than $s$.

Let us consider the following case study: $n = 500, r = 10, v = 10, s = 5, \varrho = 14, m = 6, t = 2, l_k = 16$ bytes, $l_{ID} = 2$ bytes, and $l_{kID} = 1$ byte. Moreover, in RSDTMK the final keys require an additional byte to store the permutation factor ($\mu$ bits). The memory used within the working phase is 266 bytes in QSC, is 278 bytes in RSDTMK, 186 bytes in LEAP+ and SKKE, 350 bytes in QC, 418 bytes in UKP, 200 bytes in EG, 16 bytes in PGK, 8000 bytes in FPWK. However, the cryptographic functions has not been considered, since it could be also used for the encryption, so the related memory can be saved. These dimensions can be compared with the storage area of the MSP430 microcontroller on Tmote Sky[2], which features 10 kB of RAM and 48 kB of Flash memory.

In conclusion, FPWK can be applied only to a small network, while the storage efficiency provided by the other considered protocols is comparable.

### 4.7 Communication efficiency

PPWK and PGK require the minimum quantity of messages, since all nodes know the key matched to every other node. Therefore, the key establishment is not required.

In QSC, QC, RSDTMK, LEAP+, EG, UKP and SKKE, the key establishment requires the transmission of 2 one-hop messages. However, the size of the messages is different. In QSC, the *hello* message holds the identifiers of the seeds in the ring ($r \cdot l_{sID}$) and the identifier of the node ($l_{ID}$). Even for the communication overhead, in order to provide a pessimistic analysis for the proposed approach, the quantity of key IDs transmitted with QSC is always considered the maximum ($s$). The acknowledge holds the identifiers of the sender ($l_{ID}$), the identifiers of the selected keys ($\leq s \cdot l_{kID}$), and the MAC of the message ($l_k$).

Table 6 shows the communication efficiency provided by the state-of-the-art protocols and QSC. Considering $l_k = 16$ byte, $l_{ID} = 2$ bytes, $l_{sID} = 1$ bytes, $r = 10$, $s = 5$, $t = 2$, $m = 6$ and $l_{kID} = 1$ byte (2 in RSDTMK, due to the $\mu$ bits of the permutation factor, and in UKP, which uses a pool with more than 256 keys), the pairwise key establishment requires 32 bytes in RSDTMK, 20 bytes in LEAP+, 70 bytes in SKKE, 40 bytes in QC, 56 bytes in UKP, 35 bytes in QSC, while 31 bytes in EG.

### 4.8 Analysis validation

In order to validate the proposed formulas that describe resilience and connectivity of the schemes based on random distribution, a simulative analysis has been conducted. An in-house simulator was developed in C. All tests were executed on 20 random cases per formula by iterating $10^7$ times the simulations.

In order to calculate the connectivity, two sets of $r$ numbers between 0 and $p-1$ are randomly generated. Each set represents the ring of starting keys of a node. If the two sets share at least $q$ numbers (1 for EG), the nodes can be connected.

In order to calculate the probability of a false authentication, a set of $r$ numbers between 0 and $p-1$ have been randomly generated. Then, $x$ sets of $r$ numbers, representing the compromised sets, have been generated. If at least $q$ numbers (1 for EG) of the first set are included in the $x$ sets, the adversary can pass the authentication check.

In order to calculate the probability of eavesdropping on a link, two sets of $r$ numbers between 0 and $p-1$ have been randomly generated. This operation has been iterated until the two sets share at least $q$ numbers (1 for EG). For QSC a subset of at most $s$ numbers has been selected. The selected shared numbers represent the starting keys used for a link in the network. Then, $x$ sets of $r$ numbers, representing the compromised sets, have been generated. If all numbers selected among the shared ones are included in the $x$ sets, the adversary can eavesdrop on the link.

The result of the simulative analysis confirms the validity of the proposed formulas with an average relative error equal to $10^{-3}$.

## 5 Discussion and Comparison

In this section, QSC is compared to state-of-the-art schemes. In order to obtain a quantitative comparison of the schemes, the following case study has been adopted. A network with 500 nodes, where each node has 10 neighbors, is considered. The key length is set to 128 bits, the node IDs to 16 bits and the key IDs to 8 bits. For QC, RSDTMK and EG the best $r$ and $p$ (for UKP, $m$ and $t$) are selected so that the memory storage is less than 5% of the RAM memory of Tmotes Sky (512 bytes). A connectivity higher than or equal to 0.99 is set as a constraint for the selection of suitable parameters. Since the connectivity increases with high values of $r$, but $r$ affects the storage, the highest value of $r$ compliant with the storage constraint has been selected. Then, the highest

2. Mote Sky IV, manufactured by Moteiv Corporation, http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf

TABLE 7
Parameters used for the comparison

| Parameter | Value |
|---|---|
| $n$ for UKP | 21 |
| $n$ for FPWK | 31 |
| $n$ for other schemes | 500 |
| $v$ | 10 |
| $l_k$ | 128 bits |
| $l_{ID}$ | 16 bits |
| $l_{kID}$ | 8 bits |
| max memory storage | 512 bytes |
| min connectivity | 0.99 |

TABLE 8
Values of $r$ and $p$ used for the comparison

| Protocol | $r$ | $p$ |
|---|---|---|
| EG (1-1C) | 28 | 198 |
| 1C | 19 | 97 |
| 2C | 19 | 70 |
| 3C | 19 | 57 |
| 4C | 19 | 49 |
| 1-2C | 26 | 173 |
| 1-4C | 25 | 160 |
| 1-5C | 25 | 160 |
| 1-6C | 24 | 149 |
| 1-7C | 23 | 138 |
| 1-8C | 23 | 138 |
| 1-10C | 22 | 127 |
| 1-16C | 18 | 88 |
| 2-5C | 25 | 115 |
| 3-5C | 25 | 93 |
| 4-5C | 25 | 80 |
| RSDTMK | 24 | 127 |

value of $p$ compliant with the connectivity constraint has been selected. In RSDTMK, a high value of $\mu$ improves the resilience, but it increases the memory storage. In order to provide a high level of resilience with limited drawbacks, $\mu$ is set to 8. Table 7 shows the summary of the parameters, while Table 8 shows the resulting $p$ and $r$. It can be observed that 8 bits for the key IDs are enough, since for any scheme $p < 256$. As shown in Fig. 4, the connectivity threshold strongly limits the eligible values for $m$ and $t$. According to the connectivity and memory size constraints of the proposed case study and to the characteristics of UKP presented in Section 2.1, a configuration of UKP complaint with 500 nodes does not exist, so the eligible configuration compliant with the largest possible network ($n = 21$) is selected as $m = 3$ and $t = 3$.

### 5.1 QC parameters

As shown in Table 8, $r = 19$ in QC. This value is lower than in EG, since QC requires more memory. Since the connectivity in QC is decreased by $q$, in order to reach a connectivity higher than or equal to 0.99, $p$ decreases as $q$ increases. Since for QC the level of resilience against false authentication is equal to the level of connectivity (as described in Section 4.5), in the considered case study, an adversary with one compromised node has a probability equal to 0.99 of passing an authentication check. Therefore, in order to evaluate the effects of $q$, the resilience against eavesdropping is considered. Fig. 5 shows the probability of

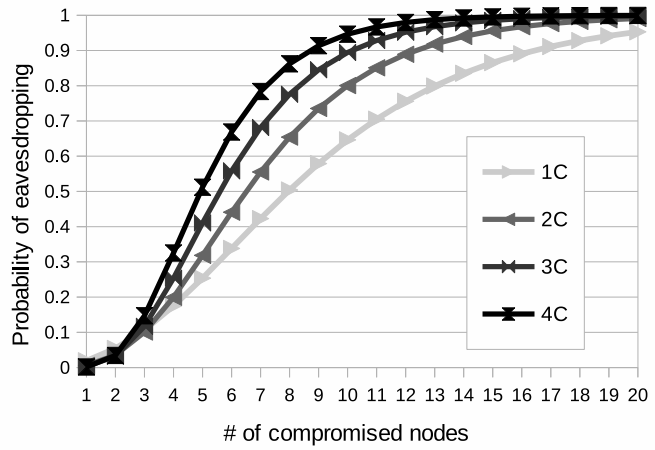


Fig. 5. Probability of eavesdropping on a link in QC, according to the number of compromised nodes.

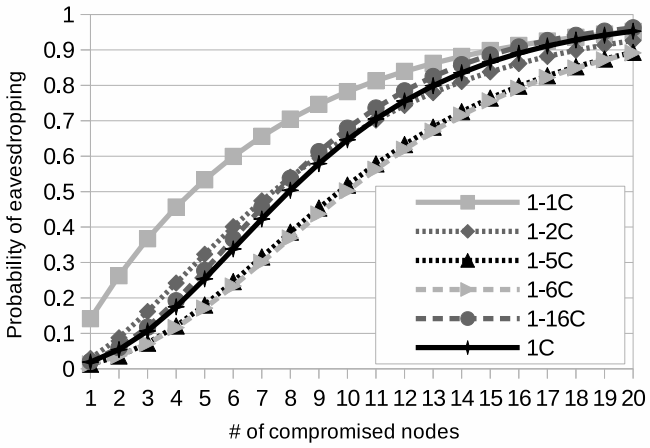


Fig. 6. Probability of eavesdropping on a link in 1-SC, according to the number of compromised nodes.

eavesdropping on a link. Although for a very low quantity of nodes a higher $q$ provides a slightly better resilience, by increasing the quantity of compromised nodes, $q = 1$ represents the best solution. This performance is due to the larger value of $p$ in 1C, which allows this configuration to provide a higher level of resilience.

### 5.2 QSC parameters

Since $q = 1$ provides the best resilience in QC, and the formula for the connectivity in QC and QSC is the same, $q = 1$ is set as the initial configuration for the analysis of QSC. Even for QSC the resilience against false authentication is equal to the connectivity. Therefore, only the resilience against eavesdropping is analyzed. Fig. 6 shows the probability of eavesdropping on a link with 1C and 1-SC. When $s = 1$, the resilience provided by 1-1C is the worst. At $s = 16$ the resilience is close to 1C. In the considered case study, $s = 5$ and $s = 6$ provide the best resilience. The best value of $s$ must be selected according to the parameters of the network. The resilience is initially improved by increasing $s$ (e.g., $s = 2$ in Fig. 6), but after a maximum (i.e., $s = 5$ in Fig. 6), it decreases again (e.g., $s = 16$ in Fig. 6). In order to

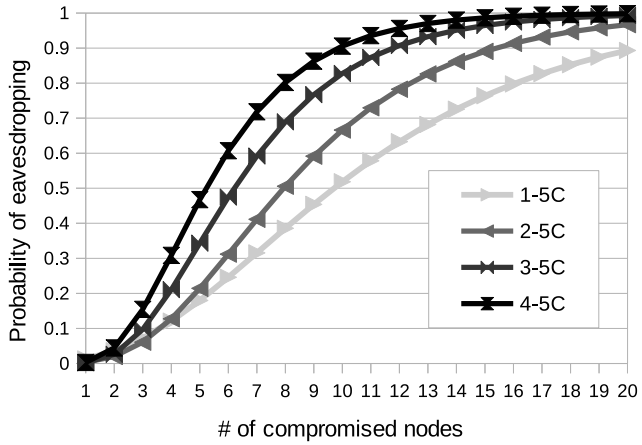


Fig. 7. Probability of eavesdropping on a link in Q-5C, according to the number of compromised nodes.

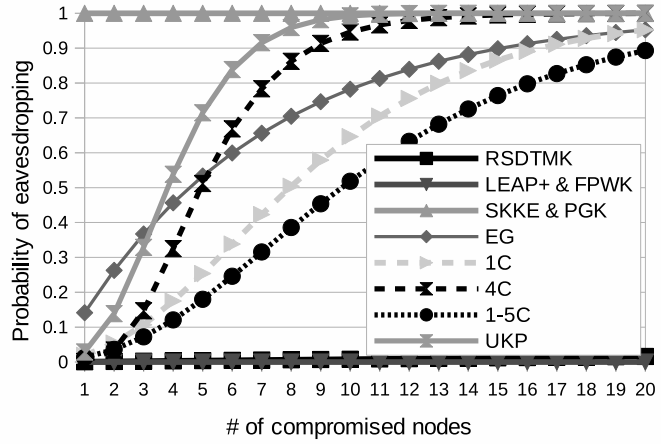


Fig. 9. Probability of eavesdropping on a link, according to the number of nodes compromised in the working phase.

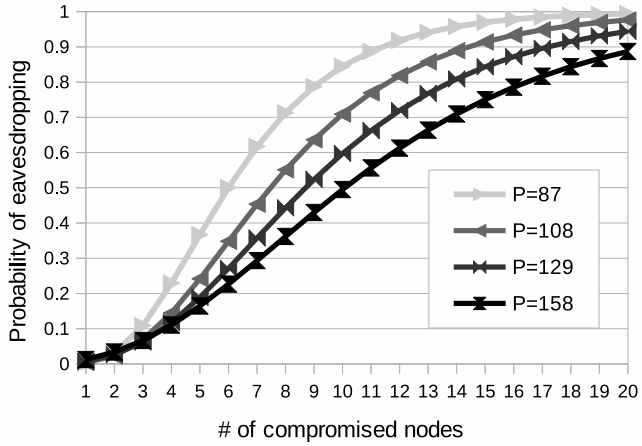


Fig. 8. Probability of eavesdropping on a link in 1-5C with various $p$, according to the number of compromised nodes.

TABLE 9
Assumptions

| Schemes | Assumptions | | |
|---|---|---|---|
| | Mobile network | Possibility of node adding | Network size unlimited |
| QSC, QC , EG, PGK & SKKE | YES | YES | YES |
| LEAP+ & RSDTMK | NO | YES | YES |
| UKP | YES | YES | NO |
| FPWK | YES | NO | NO |

check if $q = 1$ is the best configuration also for QSC, it is possible to check the values of $p$ for $s = 5$. By increasing $q$, the value of $p$ sharply decreases. Fig. 7 shows the resilience against eavesdropping according to $q$. Therefore, also for QSC the best configuration requires $q = 1$.

As observed in Section 4.4, in QSC and QC, there exist points of local maximum for the resilience against eavesdropping. Therefore, the adopted method of selection of $p$ and $r$ could not provide the best resilience complaint with the requirements of the proposed case study. In order to verify this claim, 1-5C has been analyzed. As supposed, $p = 25$ represents the best value among the eligible ones. If the quantity of compromised nodes is higher than 4, the local maximum is always higher than $p = 160$, which represents the best eligible value. If the quantity of compromised nodes is equal to 4 or less, the point of local maximum is lower than $p = 160$. Fig. 8 shows the resilience provided by four values of $p$ that correspond to the best solution respectively for one (if $x = 1$, $p = 87$), for two (if $x = 2$, $p = 108$), for three (if $x = 3$, $p = 129$), and for four nodes compromised (if $x = 4$, $p = 158$). These data were obtained by an exhaustive analysis. The improvement provided by a lower $p$ is limited. Therefore, the method adopted for the selection of the parameter $p$ and $r$ is still valid, since the selection of a lower $r$ would be counterproductive, while the selection of a lower $p$ will slightly improve the resilience for few compromised nodes, but it will consistently decrease the resilience for a higher quantity of compromised nodes.

### 5.3 Overall comparison

It is important to remark that the analyzed schemes have different requirements. As shown in Table 9, RSDTMK and LEAP+ assume that the network is static, while QSC, UKP, QC, EG, SKKE, PGK and FPWK can also be applied to mobile networks. Moreover, according to the memory requirement of the proposed case study, FPWK could be applied only to networks composed by 31 nodes at most. Moreover, FPWK does not allow adding nodes, since all nodes must be known at the deployment. About the computational and communication overheads, which are not considered in the case study, it is observed that, although some protocols are lighter, there is no significant difference that could affect the network.

Fig. 9 shows the resilience against eavesdropping provided by 1-5C and the state-of-the-art schemes if $x$ nodes are compromised in the working phase. The best resilience is provided by LEAP+ and FPWK, and it corresponds to the theoretical maximum. Even the resilience provided by RSDTMK is close to the maximum. Among the schemes without special assumptions, 1-5C provides the best resilience. 1C is generally better than EG and than UKP, while, with an higher value of $q$, QC is better than EG only for a low number of compromised nodes, in general accordance to the

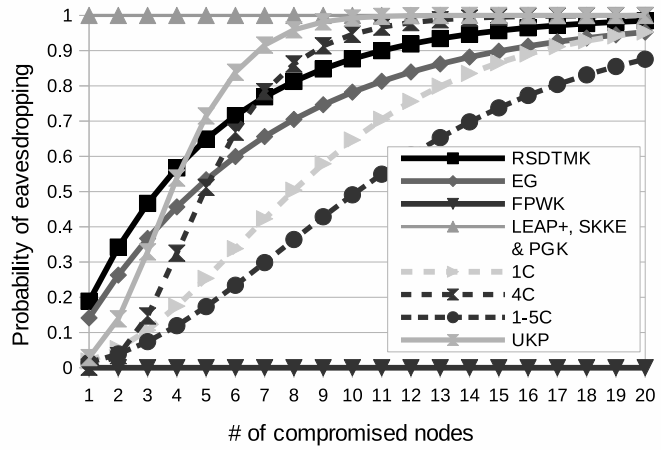


Fig. 10. Probability of eavesdropping on a link, according to the number of nodes compromised within the initialization phase.

TABLE 10
Approximated probability of passing an authentication check if at least one node has been compromised

| Schemes | Phase of the Attack | |
|---|---|---|
| | Working | Initialization |
| PGK & SKKE | 1 | 1 |
| QSC, UKP, QC & EG | $\geq$0.99 | $\geq$0.99 |
| RSDTMK | $\geq$0.0007 | $\geq$0.99 |
| LEAP+ | 0 | 1 |
| FPWK | 0 | 0 |

results presented in [15]. The worst resilience is provided by SKKE and PGK, and it corresponds to the theoretical minimum.

Fig. 10 shows the resilience against eavesdropping if $x$ nodes are compromised in the initialization phase. The only schemes in the comparison that have an initialization phase are LEAP+ and RSDTMK. The other protocols provide the same resilience. RSDTMK provides a level of resilience similar to EG. LEAP+ provides the worst resilience, like SKKE and PGK. FPWK is the only scheme that can reach the theoretical maximum.

Table 10 shows the approximated resilience against false authentication. The values are approximated in order to reach a simple representation. Therefore, the schemes that provide similar results are grouped. Although the resilience can decrease according to the quantity of compromised nodes, it is always equal or close to 0 or to 1. All protocols without special assumptions provide a probability of false authentication higher than 0.99. In particular, SKKE and PGK provide a probability equal to 1, independently of the quantity of compromised nodes. In EG, UKP, QC and QSC, the formula of the connectivity is the same as the false authentication with one compromised node. Therefore, according to the case study, they provide a probability close to 0.99 when one node is compromised, and a higher probability when the quantity of compromised node increases. If the nodes are compromised during the working phase, LEAP+ and FPWK provide a probability of false authentication equal to 0, while RSDTMK provides a probability close to 0. If the attack is performed within the initialization phase, only FPWK provides a probability of false authentication equal to 0, while LEAP+ and RSDTMK provide a probability higher than 0.99.

The best resilience is provided by FPWK. However, it has the strictest assumptions. In particular, in the case study, it could be applied only if the quantity of nodes in the network is 31 or less. Therefore, FPWK represents the best solution only for small WSNs without node adding.

For small static WSNs with node adding and for large static WSNs, if it is assumed that a node cannot be compromised during the initialization phase, LEAP+ is the best solution. If it is assumed that nodes can be compromised even during the initialization phase, 1-SC represents the most resilient scheme. RSDTMK could represent a good compromise if it is assumed that an adversary can compromise a node during the initialization phase, but that this attack represents a hard task.

For small mobile WSNs with node adding and for large mobile WSNs, 1-SC represents always the best solution.

## 6 Conclusions

In this paper, a new key distribution scheme for wireless sensor networks, called $q$-$s$-composite, has been proposed. It is based on random predistribution of the secret material. The main benefit of $q$-$s$-composite is represented by an efficient memory management, which allows to store a larger quantity of keys and consequently it can improve the resilience of the protocol. This result is reached by means of a new key generation mechanism and by limiting the quantity of starting keys per link. The potential drawbacks of the proposed scheme have been analyzed and an in-depth analysis has shown that their effects are overcome by the security improvements.

A comparison with state-of-the-art schemes shows that the proposed approach represents the best solution for large mobile WSNs, and that it is also the best solution for static WSNs, if the nodes can be compromised during the initialization phase.

## Acknowledgment

This work was partially supported by the grant "Bando Smart Cities and Communities", OPLON Project (OPportunities for active and healthy LONgevity) funded by the Italian Ministry of University.

## References

[1] H. Dai, Z. min Zhu, and X.-F. Gu, "Multi-target indoor localization and tracking on video monitoring system in a wireless sensor network," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 228 – 234, 2013.

[2] J. Gisbert, C. Palau, M. Uriarte, G. Prieto, J. Palazn, M. Esteve, O. Lpez, J. Correas, M. Lucas-Esta, P. Gimnez, A. Moyano, L. Collantes, J. Gozlvez, B. Molina, O. Lzaro, and A. Gonzlez, "Integrated system for control and monitoring industrial wireless networks for labor risk prevention," *Journal of Network and Computer Applications*, vol. 39, pp. 233 – 252, 2014.

[3] G. Hackmann, W. Guo, G. Yan, Z. Sun, C. Lu, and S. Dyke, "Cyber-physical codesign of distributed structural health monitoring with wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 1, pp. 63–72, Jan 2014.

[4] A. A. Rezaee, M. H. Yaghmaee, A. M. Rahmani, and A. H. Mohajerzadeh, "Hoca: Healthcare aware optimized congestion avoidance and control protocol for wireless sensor networks," *Journal of Network and Computer Applications*, vol. 37, pp. 216 – 228, 2014.

[5] X. Du and H.-H. Chen, "Security in wireless sensor networks," *Wireless Communications, IEEE*, vol. 15, no. 4, pp. 60–66, Aug 2008.

[6] M. A. Simplício-Jr., P. S. Barreto, C. B. Margi, and T. C. Carvalho, "A survey on key management mechanisms for distributed wireless sensor networks," *Computer Networks*, vol. 54, no. 15, pp. 2591 – 2612, 2010.

[7] H. Moosavi and F. Bui, "A game-theoretic framework for robust optimal intrusion detection in wireless sensor networks," *Information Forensics and Security, IEEE Transactions on*, vol. 9, no. 9, pp. 1367–1379, Sept 2014.

[8] S. Shamshirband, N. B. Anuar, M. L. M. Kiah, V. A. Rohani, D. Petkovi, S. Misra, and A. N. Khan, "Co-fais: Cooperative fuzzy artificial immune system for detecting intrusion in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 42, no. 0, pp. 102 – 117, 2014.

[9] C.-M. Yu, Y.-T. Tsou, C.-S. Lu, and S.-Y. Kuo, "Localized algorithms for detection of node replication attacks in mobile sensor networks," *Information Forensics and Security, IEEE Transactions on*, vol. 8, no. 5, pp. 754–768, May 2013.

[10] S.-H. Seo, J. Won, S. Sultana, and E. Bertino, "Effective key management in dynamic wireless sensor networks," *Information Forensics and Security, IEEE Transactions on*, vol. 10, no. 2, pp. 371–383, Feb 2015.

[11] X. Du, M. Guizani, Y. Xiao, and H.-H. Chen, "Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 3, pp. 1223–1229, March 2009.

[12] B. Zhou, S. Li, Q. Li, X. Sun, and X. Wang, "An efficient and scalable pairwise key pre-distribution scheme for sensor networks using deployment knowledge," *Computer Communications*, vol. 32, no. 1, pp. 124–133, 2009.

[13] T. Kwon, J. H. Lee, and J. Song, "Location-based pairwise key predistribution for wireless sensor networks," *Wireless Communications, IEEE Transactions on*, vol. 8, no. 11, pp. 5436–5442, November 2009.

[14] L. Eschenauer and V. Gligor, "A key-management scheme for distributed sensor networks," in *Computer and communications security: CCS '02, 9th ACM conf. on*, 2002, pp. 41–47.

[15] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks," in *Symposium on Security and Privacy.*, May 2003, pp. 197–213.

[16] 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.

[17] W. Bechkit, Y. Challal, A. Bouabdallah, and V. Tarokh, "A highly scalable key pre-distribution scheme for wireless sensor networks," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 2, pp. 948–959, February 2013.

[18] A. K. Das, "A random key establishment scheme for multi-phase deployment in large-scale distributed sensor networks," *International Journal of Information Security*, vol. 11, no. 3, pp. 189–211, 2012.

[19] S. Zhu and et al, "Leap+: Efficient security mechanisms for large-scale distributed sensor networks," *ACM Transactions on Sensor Networks*, vol. 2, no. 4, pp. 500–528, 2006.

[20] F. Gandino, B. Montrucchio, and M. Rebaudengo, "Key management for static wireless sensor networks with node adding," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 2, pp. 1133–1143, May 2014.

[21] A. K. Das, "An efficient random key distribution scheme for large-scale distributed sensor networks," *Security and Communication Networks*, vol. 4, no. 2, pp. 162–180, 2011.

[22] R. Blom, "An optimal class of symmetric key generation systems," in *EUROCRYPT 84 workshop on Advances in cryptology: theory and application of cryptographic techniques*. New York, NY, USA: Springer-Verlag, 1985, pp. 335–338.

[23] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 2, pp. 228–258, 2005.

[24] D. Liu, P. Ning, and R. Li, "Establishing pairwise keys in distributed sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 1, pp. 41–77, 2005.

[25] D. Liu and P. Ning, "Improving key predistribution with deployment knowledge in static sensor networks," *ACM Trans. Sen. Netw.*, vol. 1, no. 2, pp. 204–239, 2005.

[26] Z. Yu and Y. Guan, "A key management scheme using deployment knowledge for wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 19, no. 10, pp. 1411–1425, Oct. 2008.

[27] M. Younis, K. Ghumman, and M. Eltoweissy, "Location-aware combinatorial key management scheme for clustered sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 17, no. 8, pp. 865–882, Aug 2006.

[28] K. Ren, W. Lou, and Y. Zhang, "Leds: Providing location-aware end-to-end data security in wireless sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 7, no. 5, pp. 585–598, May 2008.

[29] Q. Dong and D. Liu, "Using auxiliary sensors for pairwise key establishment in wsn," *ACM Trans. Embed. Comput. Syst.*, vol. 11, no. 3, pp. 59:1–59:31, Sept. 2012.

[30] A. K. Das, "Improving identity-based random key establishment scheme for large-scale hierarchical wireless sensor networks." *IJ Network Security*, vol. 14, no. 1, pp. 1–21, 2012.

[31] M. Eltoweissy, M. Moharrum, and R. Mukkamala, "Dynamic key management in sensor networks," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 122–130, April 2006.

[32] *NIST FIPS PUB 197: Advanced Encryption Standard (AES)*, Federal Information Processing Standards Publication 197, US Department of Commerce/N.I.S.T. Std., November 2001.

[33] J. Sun and X. Zhang, "Study of zigbee wireless mesh networks," in *Hybrid Intelligent Systems, 2009. HIS '09. Ninth International Conference on*, vol. 2, Aug 2009, pp. 264–267.

[34] D. H. Yum and P. J. Lee, "Exact formulae for resilience in random key predistribution schemes," *Wireless Communications, IEEE Transactions on*, vol. 11, no. 5, pp. 1638–1642, May 2012.

**Filippo Gandino** obtained his M.S. in 2005 and Ph.D. degree in Computer Engineering in 2010, from the Politecnico di Torino. He is currently an Assistant Professor with the Dipartimento di Automatica e Informatica, Politecnico di Torino. His research interests include ubiquitous computing, RFID, WSNs, security and privacy, network modeling and digital arithmetic.

**Renato Ferrero** received the M.S. degree in computer engineering in 2004 and the Ph.D. degree in Computer Engineering in 2012, both from Politecnico di Torino, Italy. He is currently a research fellow at the Dipartimento di Automatica e Informatica of Politecnico di Torino. His research interests include ubiquitous computing, wireless sensor networks and RFID systems.

**Maurizio Rebaudengo** received the MS degree in Electronics (1991), and the PhD degree in Computer Engineering (1995), both from Politecnico di Torino, Italy. Currently, he is an Associate Professor at the Dipartimento di Automatica e Informatica of the same institution. His research interests include ubiquitous computing and testing and dependability analysis of computer-based systems.