Computers, Materials & Continua CMC, vol.64, no.2, pp.1261-1280, 2020

PUF-Based Key Distribution in Wireless Sensor Networks

Zheng Zhang[1], Yanan Liu[1, *], Qinyuan Zuo[1], Lein Harn[2], Shuo Qiu[1] and Yuan Cheng[1]

Abstract: Physical Unclonable Functions (PUFs) can be seen as kind of hardware one-way functions, who are easily fabricated but difficult to clone, duplicate or predict. Therefore, PUFs with unclonable and unpredictable properties are welcome to be applied in designing lightweight cryptography protocols. In this paper, a Basic Key Distribution Scheme (Basic-KDS) based on PUFs is firstly proposed. Then, by employing different deployment modes, a Random Deployment Key Distribution Scheme (RD-KDS) and a Grouping Deployment Key Distribution Scheme (GD-KDS) are further proposed based on the Basic-KDS for large scale wireless sensor networks. In our proposals, a sensor is not pre-distributed with any keys but will generate one by the embedded PUF when receiving a challenge from the gateway, which provides perfect resilience against sensor capture attacks. Besides, the unclonable and unpredictable properties of PUF guarantee the key uniqueness and two-way authentication. Analysis and experiment results show that our proposals have better performances in improving the resilience, secure-connectivity, and efficiency as compared to other schemes.

Keywords: Key distribution, physical unclonable functions, PUF, wireless sensor networks, deployment mode.

1 Introduction

Wireless Sensor Networks (WSNs), deployed in hostile environments, are prone to various types of malicious attacks, including physical capture of a node, intentionally providing misleading information, impersonation, data modification, and so on [Carman, Kruus and Matt (2000); Chen and Chao (2011)]. Key distribution and authentication are basis and pre-condition of security communication [Das, Zeadally and He (2018)]. A large scale WSN is often clustered constructed [Bohge and Trappe (2003)] and consists of different nodes, e.g., sensors, gateways, and a server, as shown in Fig. 1. In a cluster, the gateway is a data processing and fusing center; it transmits the aggregated data to the server via long-haul communication [Wang, Shao, Gao et al. (2019)]. A large quantity of sensors are divided into non-overlapping clusters; they collect information from environment and send the raw data to

[1] Jinling Institute of Technology, Nanjing, 211169, China.

[2] Department of Computer Science Electrical Engineering, University of Missouri, Kansas, 64110, USA.

[*] Corresponding Author: Yanan Liu. Email: yanan.liu@jit.edu.cn.

Received: 05 February 2020; Accepted: 16 April 2020.

gateways via short-haul communication. Compared with sensors, gateways usually have more power and memory. Many well-known secure key management methods, like Diffie-Hellman scheme, public cryptography, can be employed between gateways or a gateway and a server. However, sensors always have limited resources in memory, computation, and communication. Therefore, the public key cryptography [Kadri, Feham and Mhammed (2012)], identity-based cryptography [Ge, Liu, Xia et al. (2019)], or quantum key distribution [Zhang, Chang, Yan et al. (2019)] are not appropriate for such low-ended sensors because of the huge resource demand. Moreover, in literal key distribution schemes [Du, Xiao, Guizani et al. (2007); Boujelben, Cheikhrouhou, Abid et al. (2009); Zhang and Varadharajan (2010); Eschenauer and Gligor (2003)] of recent years, sensors are always pre-distributed with some keys so as to establish pairwise keys with neighbors (including gateways). This kind of so-called key pre-distribution schemes save the computation and communication overheads, but cannot resist node physical capture attacks, which threaten the whole system failure when the pre-distributed keys are extracted by attackers.

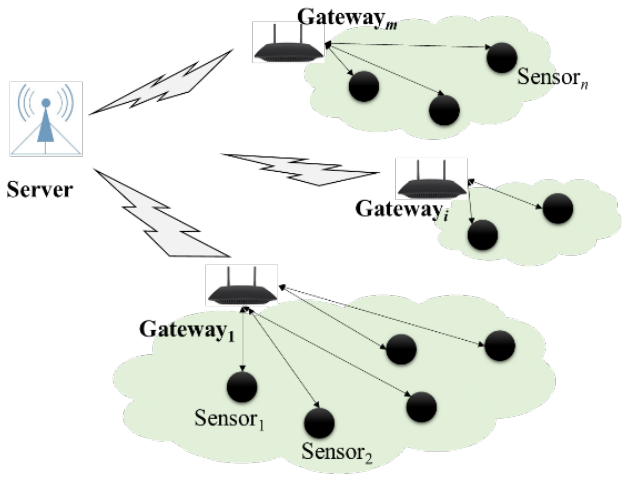


Figure 1: Wireless sensor network model

PUFs are kind of an integrated circuit (IC) that transforms the physical disorder of random semiconductor fabrication process variations of its nanoscale devices into a unique and unpredictable digital bit-stream (response) upon query (challenge) [Pappu, Recht and Gershenfeld (2002)]. They can be seen as the hardware equivalent of a one-way function, who are easy to fabricate but practically infeasible to clone, duplicate or predict, even if the exact manufacturing process is produced again. In this regard, PUFs with unclonable and unpredictable properties are increasingly becoming a vital security tool [Mukhopadhyay (2016); Rahman, Rahman, Forte et al. (2016)] in the wireless sensor network [Mao, Zhang, Qi et al. (2019)], Internet of Things (IoT) [Ren, Zhu, Sharma et al. (2020); Fang, Li, Yun et al. (2019)], and cloud applications [Li, Yan, Chen et al. (2019)], especially in device authentication and key generation schemes [Aman and Chua (2016)]. Instead of using digital certificate, the authentication of PUFs is based on the usage of Challenge-Response Pairs (CRPs).

The contributions we made in this paper are:

Firstly, we propose a Basic-KDS for a gateway to distribute a session key to a sensor. The sensor is not pre-loaded with any key in the memory, but will generate a PUF response on-the-fly when receiving a challenge, which is used to decrypt the session key distributed from the gateway. Therefore, it is perfectly resilient against sensor capture attacks. Besides, the two-way authentication between a gateway and a sensor is also guaranteed based on the PUF CRPs.

Secondly, we combine the Basic-KDS with a random deployment model to propose a RD-KDS for large scale sensor networks. In order to increase the secure connectivity, a sensor is designated to λ (λ>=1) gateways (denoted as parent-gateways), each of which saves a PUF CRP tuple for the sensor. If a sensor is deployed into the cluster coverage of its parent-gateway, the Basic-KDS can be implemented inside of the cluster. Proof and experiments are given to analyze the security and performances.

Thirdly, we combine the Basic-KDS with grouping deployment model [Liu, Ning and Du (2008)] to propose a GD-KDS to further increase the connectivity. A grouping model is designed, in which all nodes are divided into different deployment groups. The network destination area is also divided into non-overlapping zones. A group of nodes are deployed together into a zone so as to increase the probability of forming a cluster. Analysis and experiments prove that GD-KDS improves the secure-connectivity without increasing the storage overhead.

Fourthly, property comparisons among different key distribution schemes are also provided to prove our proposals have much improved resilience and performances.

The rest of paper is organized as follows. In Section 2, we build the PUF structure and the network model. Section 3 introduces related works. In Section 4, we propose the Basic-KDS and analyze its resilience, authentication and PUF security. RD-KDS and GD-KDS are proposed in Sections 5 and 6, analysis and experiment results are also given. Section 7 compares different key distribution schemes in resilience, secure-connectivity, and overheads. We conclude the paper with future research directions in Section 8.

2 Model building

2.1 PUF structure

PUF-based modules are utilized to generate secrets from random process variation in the fabrication of integrated circuits instead of storing the secrets in memory [Gassend, Clarke, van Dijk et al. (2002); Lee, Lim, Gassend et al. (2004)]. They do not require expensive cryptographic hardware such as the Secure Hash Algorithm (SHA) or a public/private key encryption algorithm. The “secret” is derived from physical characteristics of the Integrated Circuit (IC), therefore one cannot manufacture two identical chips, even with full knowledge of the chips design. PUF modules are being researched as an alternative that essentially leverages unique behavior of a device due to manufacturing variations as a hardware-based fingerprint. Except of IC-based PUF, there are optical PUFs [Pappu, Recht, Taylor et al. (2002)], silicon PUFs [Gassend, Clarke, Dijk et al. (2002)] and coating PUFs [Tuyls, Schrijen, Skoric et al. (2006)].

Each PUF can be modeled as a black-box challenge-response system, where a set of challenges are available and the system responds with a set of sufficiently different responses. We use a one-way mapping function to describe PUF, which can be expressed as Eq. (1).

$$\Gamma: X \rightarrow Y: \Gamma(x)=y, x \in X, y \in Y, \quad (1)$$

In Eq. (1), X and Y are challenge set and response set respectively, and Γ is a PUF function [Huang, Yu and Li (2018)] (as shown in Fig. 2). A specific challenge x and its corresponding response y together form a Challenge-Response Pair (CRP) (x, y) for the given PUF Γ.

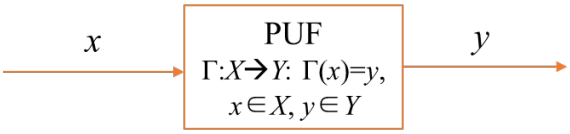


Figure 2: PUF is a one-way mapping function

Due to the low-weighted, unclonable and unpredicted properties, PUFs have two primary applications in wireless sensor networks: (1) low-cost authentication; and (2) secure key generation. A node embedded with a PUF module generates a response on-the-fly rather than saving keys in memory, which reduces not only the storage overhead but also the risk of key exposure.

Recently, a kind of sensor based wearable PUF has been introduced [Fukushima, Hidano and Kiyomoto (2016)] that utilizes similar features offered by numerous sensors available on such devices. To access such hardware intrinsic characteristics, introduction of low-level access can either be incorporated during manufacturing stage or through addition of a specific device driver. By utilizing one-way functions, characteristics of these sensors can be utilized to generate an exclusive hardware signature of the device such as a smart phone or smart watch of a smart home resident.

2.2 Assumptions

We mainly consider the node capture attack to the network. The objective of the adversary is to try to manipulate the whole network through capturing a fraction of nodes and compromising the keys in their memories. No tamper-proof is assumed on both sensors and gateways, which means that the adversary can trivially read the keys stored in the captured nodes. Moreover, the attacker can foster collusion among a group of captured nodes in order to uncover more keys in the remaining healthy network. From this point of view, capturing gateway nodes brings more damages than capturing sensors since they always store more secret information.

Assumptions. We adopt the communication assumptions in Jolly et al. [Jolly, Kuscu, Kokate et al. (2003)]: (1) the secure communications between the sensor and gateway are mainly focused in this paper; (2) the key agreement between gateways, between a gateway and the server are not involved in this paper (because many well-known solutions can be utilized, e.g., Diffie-Hellman Key Exchange, public key-based key

distribution, and etc.) (3) the end-to-end communication between sensors in a cluster is assumed to be avoided.

2.3 Notations

Parameters and notations used in this paper are given in Tab. 1.

Table 1: Notations

Notations	Description
CRP	Challenge-response pair.
$\Gamma(x)=y$	Γ is a PUF, x is the challenge, y is the response.
<id, x, y>	A tuple including the node id and a PUF CRP.
$cipher=E(key, plain)$	Encrypt the plain with the key and get the cipher.
$plain=D(key, cipher)$	Decrypt the cipher with the key and get the plain.
λ	Number of gateways loading the CRP tuple of a sensor.
l	The length of the rectangle destination area.
w	The width of the rectangle destination area.
m	The number of gateways.
n	The number of sensors.
r	The circle radius of the gateway's coverage area.
f_c	The fraction that a sensor in a cluster formed by its parent gateway.
p_t	The fraction that a gateway loaded with a sensor's CRP.
P	The secure-connectivity.
t_a	The average amount of tuples saved in a gateway.
n_a	The average amount of sensors in a cluster.
Di_Ov	The overhead of a gateway in Direct Key Distribution.
In_Ov	The overhead of a gateway in Indirect Key Distribution.
DG_i	Deployment group with the index of i.
$F_r(x)$	The resilience against x nodes are captured.

3 Related works

In 2003, Jolly et al. [Jolly, Kuscu, Kokate et al. (2003)] proposed a Low-Energy Key Management (LEKM) protocol in sensor networks. It was the minimization of the sensors' energy consumption. However, there were two restrictions in the LEKM

protocol. First, the key revocations/renewals operations were too much dependent on the server, which was uneconomical and vulnerable to attacks. Second, when some gateways were captured by attackers, the sensors in their clusters became "orphaned sensors" and they could not re-establish secure links with healthy gateway nodes.

In 2007, Du et al. [Du, Xiao, Guizani et al. (2007)] proposed an asymmetric pre-distribution scheme, which employed the probabilistic scheme [Eschenauer and Gligor (2003)] into clustered WSNs. However, in this scheme, an attacker could compromise a large number of keys by capturing a small fraction of sensors or gateway nodes, then the attacker could take control the entire network by eavesdropping the communications between un-captured nodes, or deploying a replicated nodes loaded with some compromised keys. In 2009, Boujelben et al. [Boujelben, Cheikhrouhou, Abid et al. (2009)] proposed a pairwise key management scheme based on Blom's matrix scheme [Blom (1985)], which improved the resilience against node capture but consumed too much storage overhead of sensors to meet a certain security and connectivity requirements. All of the works mentioned above rely on symmetric key cryptography.

Public key cryptography is also used to securely bootstrap the pairwise key over a public communication channel in WSNs. For example, in 2012, Alagheband et al. [Alagheband and Aref (2012)] proposed a dynamic and secure key management model for hierarchical heterogeneous sensor networks and Kadri et al. [Kadri, Feham and Mhammed (2012)] proposed an architecture aware key management scheme for wireless sensor networks. These two works were proposed based on the Elliptical Curve Cryptography (ECC). Key agreement between a gateway and a sensor is implemented by involving the server, which brought a huge increase of the communication overhead. Besides, although resource-constrained sensor nodes are able to use Public Key Cryptography (PKC) through ECC, the amount of memory required to implement the algorithm and the time/energy needed to complete the negotiation is substantially higher than symmetric cryptography.

PUF-based key generation schemes are suggested in recent works. Guajardo et al. [Guajardo, Kumar and Tuyls (2008)] presented two protocols for secure key deployment without any pre-shared key. Their protocols take advantage of PUF characteristics to distribute shared key between nodes, one uses a trusted third party (TTP) and another one does not. However, the CRPs are sent in plain text over insecure channels between parties, which gives chances to the attacker to build the CRP related to PUF devices of nodes to apply an impersonation attack.

Bahrampour et al. [Bahrampour and Atani (2013)] proposed a key management scheme for WSNs that was secure against node capture attacks. This scheme used PUF at each node and pre-distributed public key pairs over all nodes in the network. However, this scheme depended on the public key algorithm, which was more computationally complex than the symmetric algorithm.

4 Basic key distribution scheme (Basic-KDS)

4.1 Proposal

We use a basic scheme to explain how to employ the PUF to establish a session key between a sensor S and its gateway G without pre-distributed keys.

4.1.1 Step 1: Initialization

Before network deployment, each sensor S is embedded with a PUF, denoted as Γ_S. Take a random challenge number C as the input of Γ_S and get the output response R. Save the tuple $<id_S, C, R>$ into the gateway G.

4.1.2 Step 2: Key Distribution

After network deployment, the gateway G reads the challenge C from database and sends it to sensor S. The sensor takes the challenge C as PUF input and gets the corresponding response R as Eq. (2).

$$R=\Gamma_S(C). \tag{2}$$

The sensor uses R to encrypt C and sends the *cipher* to the gateway:

$$cipher=E(R, C). \tag{3}$$

In Eq. (3), R is the encryption key, C is the plaintext, and *cipher* is the ciphertext. The gateway uses R to decrypt the *cipher* and compares the *plain* with the C:

$$plain=D(R, cipher). \tag{4}$$

In Eq. (4), R is the decryption key, *cipher* is received from S, and *plain* is the decryption result.

If $plain=C$ is true, the gateway believes the sensor S is trusted and authenticated during the *Initialization* step. Then it generates a session key k_{GS} from a Key Generation Module (KGM) and sends it to the sensor S by encrypted with R, as Eq. (5):

$$cipher2=E(R, k_{GS}). \tag{5}$$

The sensor S decrypts *cipher*2 and gets k_{GS}.

Otherwise, if $plain=C$ is false, the gateway considers the *cipher* is coming from an un-authenticated sensor and will not establish session key with such a sensor.

The key distribution process in Basic-KDS is described in Fig. 3.

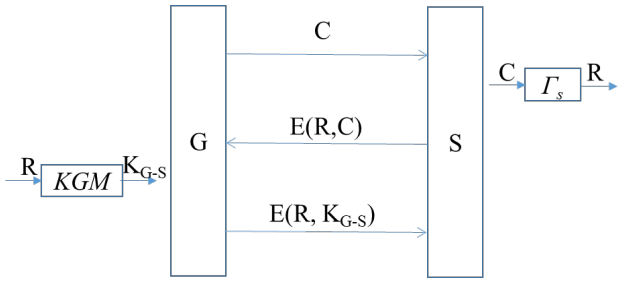


Figure 3: Key distribution in Basic-KDS

In this Basic-KDS, the sensors are not pre-loaded with any keys during the *Initialization* step, which is perfectly resilient against node capture attacks, because the compromised nodes will not disclose any links between un-compromised nodes. Besides, the transmission process is concise and the calculation on PUF chip is light-weighted.

4.2 Authentication

In the Basic-KDS, the two-way authentication between a gateway and a sensor is provided based on the PUF CRP. The gateway saves the CRP tuple <*id*, *C*, *R*> of each authorized sensor, who is embedded with a PUF. When receiving a challenge *C*, the sensor generates a corresponding response *R* by the PUF and sends the ciphertext *cipher*=*E* (*R*, *C*) to the gateway. The gateway decrypts the ciphertext and checks whether the *plain*=*D* (*R*, *cipher*) is equal to the challenge *C*. These operations help the gateway to authenticate the sensor, because only authorized sensor can supply the correct response *R* by receiving the challenge *C*. If the *plain* is not equal to the challenge *C*, the gateway considers the sensor is not authorized because it does not possess the correct PUF. Then the gateway sends an encrypted session key k_{GS} by using *R* as the key. This operation helps the sensor to authenticate the gateway, because only authorized gateway is acquaint with the correct PUF CRP tuple of the sensor.

4.3 Resilience against node capture attack

The unattended operation of sensors and no tamper-proof design raises the success rate of node capture attack by adversaries. By reading the keys or other secret information in a fraction of captured nodes' memory, the adversaries could further foster collusion attacks to the whole network.

In Eq. (6), we use the fraction of compromised links between un-captured nodes $F_r(x_c)$ to define the resilience against node capture, where x_c is the number of captured nodes.

$$F_r(x_c) = \frac{number\ of\ compromised\ links\ between\ uncaptured\ nodes}{number\ of\ all\ links\ between\ uncaptured\ nodes} \tag{6}$$

4.3.1 Resilience against sensor capture attack

Our key distribution schemes are proposed based on the PUF, in which, sensors are embedded with a PUF instance (or chip) but not pre-distributed keys. A sensor does not store any PUF CRP, but generates a response by a received challenge on-the-fly. Therefore, we have perfect resilience against sensor capture, since the adversary cannot compromise any links of the un-captured network from a captured sensor.

$$F_r(x_c)=0. \tag{7}$$

In Eq. (7), the x_c is denoted as the number of compromised sensor nodes.

4.3.2 Resilience against gateway capture attack

We assume that the key distribution process is so quick that the adversary cannot conduct the node capture attack in such short time.

In the Basic-KDS, some PUF CRP tuples, but not keys, are saved in gateways' memory. Therefore, adversaries cannot directly achieve the session keys between uncompromised links from captured gateways:

$$F_r(x_{c\text{-}GW})=0. \tag{8}$$

In Eq. (8), the $x_{c\text{-}GW}$ is the number of captured gateways.

4.5 PUF security

The security of PUF is based on the physical differences which cannot be repeated at the industrial technology level [Eschenauer and Gligor (2002)]. PUF needs to be integrated into the chip of a node. That is to say, assume that the attacker knows PUF structure and some challenges, they still cannot achieve the corresponding responses. PUF can resist node capture, cloning and side channel attacks including electromagnetic analysis attack, differential fault attack, etc., and can completely avoid physical data leakage of nodes.

5 Random deployment key distribution scheme (RD-KDS)

We employ the Basic-KDS into a Hierarchical Sensor Network (HSN), aiming to help the gateway distribute the session keys to the sensors in its cluster.

5.1 Proposal

5.1.1 Initialization

Similar with the Basic-KDS, before network deployment, each sensor S is embedded with a PUF, denoted as Γ_s. Take a set of λ random challenges $C_{i=1,\ldots,\lambda}$ as the input of Γ_s and get the output response $R_{i=1,\ldots,\lambda}$. Select λ gateways as the parent-gateways of sensor S, and save tuples $<id_s, C_i, R_i>_{i=1,\ldots,\lambda}$ into the parent-gateway $PG_{i=1,\ldots,\lambda}$.

As shown in Fig. 4, there are 4 sensors and 3 gateways. Each sensor is designated to 2 (λ=2) gateways. For example, the sensor S_1 is designated to G_1 and G_2. Meanwhile, G_1 is also designated by sensor S_3, and G_2 is designated by sensor S_2 and S_3. In such way, we can see G_1 is loaded with 2 tuples: $<id_{S1}, C_{S1\text{-}1}, R_{S1\text{-}1}>$ and $<id_{S3}, C_{S3\text{-}1}, R_{S3\text{-}1}>$, and G_2 is loaded with 3 tuples: $<id_{S1}, C_{S1\text{-}2}, R_{S1\text{-}2}>$, $<id_{S2}, C_{S2\text{-}1}, R_{S2\text{-}1}>$, and $<id_{S4}, C_{S4\text{-}1}, R_{S4\text{-}1}>$.

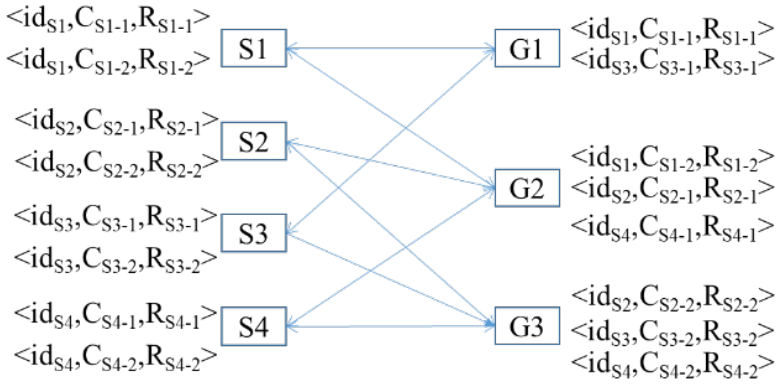


Figure 4: Many-to-many mapping between sensors and gateways

5.1.2 Step 2: key distribution

After network deployment, the gateway launches the cluster forming process firstly (the clustering algorithm is not discussed in this paper, which can be referred in Younis et al. [Younis, Youssef and Arisha (2002)]). Suppose all nodes, including gateways and sensors, are randomly deployed into the destination area of the network.

As shown in Fig. 5, the sensors S_1 and S_4 locate in the coverage area of the gateway G_1. Since it is loaded with the tuple $<id_{S1}, C_{S1\text{-}1}, R_{S1\text{-}1}>$ in the *Initialization* step, the gateway G_1 is able to directly authenticate and distribute a session key to S_1 via the Basic-KDS.

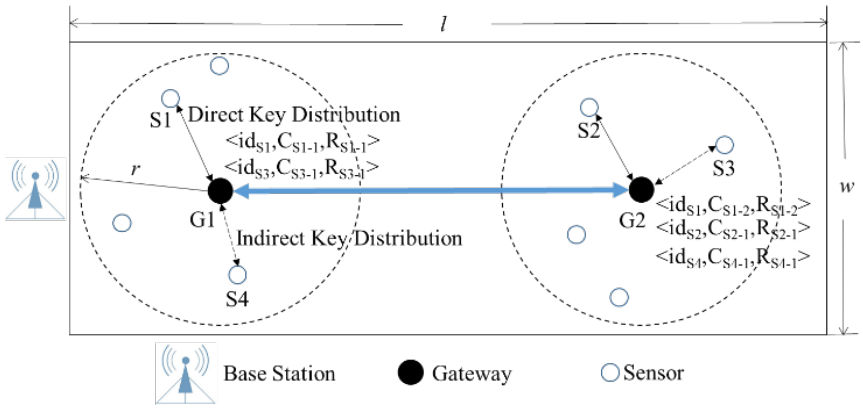


Figure 5: Node random deployment

Fig. 6 illustrate the process of Direct Key Distribution between G_1 and S_1.

(1) $G_1 \rightarrow S_1$: $C_{S1\text{-}1}$
(2) S_1: $R_{S1\text{-}1}=\Gamma_{S1}(C_{S1\text{-}1})$
(3) $S_1 \rightarrow G_1$: $cipher=E(R_{S1\text{-}1}, C_{S1\text{-}1})$
(4) G_1: $plain=D(R_{S1\text{-}1}, cipher)$
(5) If $plain = C_{S1\text{-}1}$ is true,
(6) $G_1 \rightarrow S_1$: $cipher_2=E(R_{S1\text{-}1}, K_{G1\text{-}S1})$
(7) S_1:$K_{G1\text{-}S1}=D(R_{S1\text{-}1}, cipher_2)$

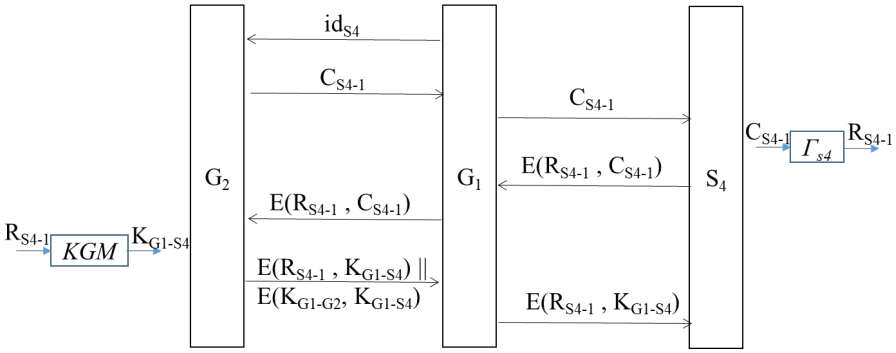
Figure 6: G_1 directly authenticates and distributes a key to S_1

5.1.3 Step 3: indirect key distribution

As shown in Fig. 5, the sensor S_4 is also located in the coverage area of the gateway G_1, but cannot be directly authenticated by G_1 without the PUF CRP of sensor S_4. In this situation, the gateways G_1 firstly broadcasts the *id* of sensor S_4 to find a neighboring gateway who has S_4's PUF CRP, e.g., G_2.

Fig. 7 illustrates the process of Indirect Key Distribution between G_1 and S_4.

Assume the communication between G_1 and G_2 is secured by a gateway-to-gateway session key $K_{G1\text{-}G2}$, which is already established by some other key agreement methods, e.g., Diffie-Hellman Key Exchange, public key-based key distribution, and etc.



(1) G_1→broadcast: id_{S4}
(2) G2→ G_1: C_{S4-1}
(3) G_1→S_4: C_{S4-1}
(4) S_4: $R_{S4-1}=\Gamma_{S4}(C_{S4-1})$
(5) S_4→ G_1: $cipher=E(R_{S4-1}, C_{S4-1})$
(6) G_1→G_2: $cipher=E(R_{S4-1}, C_{S4-1})$
(7) G_2: $plain=D(R_{S4-1}, cipher)$
(8) If $plain= C_{S4-1}$ is true,
(9) G_2→ G_1: $cipher_2=E(R_{S4-1}, K_{G1-S4}) \| cipher_3=E(K_{G1-G2}, K_{G1-S4})$
(10) G_1: $K_{G1-S4}=D(K_{G1-G2}, cipher_3)$
(11) G_1→S_4: $cipher_2$
(12) S_4: $K_{G1-S4}=D(R_{S4-1}, cipher_2)$

Figure 7: G_1 indirectly authenticates and distributes a key to S_4

5.2 Secure-connectivity

The secure-connectivity is defined as the probability that a shared key can be established between two nodes. In this paper, we re-define the secure-connectivity as the probability that a gateway can establish a pairwise key with a sensor inside its cluster.

In the proposed RD-KDS, each sensor can establish pairwise keys with its gateway by direct or in-direct steps. Therefore, the RD-KDS is a deterministic scheme which guarantees a 100% secure-connectivity, which is an improvement compared with the existing probabilistic schemes [Du, Xiao, Guizani et al. (2007); Boujelben, Cheikhrouhou, Abid et al. (2009)].

In the following, we specifically analysis the direct secure-connectivity of the RD-KDS.

Suppose the number of gateways and sensors is m and n. And suppose the network destination area is a rectangle where the length is l and the width is w. The coverage area of a gateway is a circle with the radius is r. The average amount of sensors located in the coverage area of a gateway is n_a achieved in Eq. (9).

$$n_a = \frac{n}{m}, \tag{9}$$

The average amount of tuples that are saved in a gateway is t_a achieved in Eq. (10).

$$t_a = \frac{\lambda n}{m}, \tag{10}$$

Hence, in a cluster, the fraction that the gateway is loaded with a sensor's tuple is denoted as p_t achieved in Eq. (11).

$$p_t = 1 - \frac{C_{n-t_a}^{n_a}}{C_n^{n_a}} \quad . \tag{11}$$

And the secure-connectivity that a gateway can directly authenticate and distribute key with a sensor is P in Eq. (12).

$$P = p_t \quad . \tag{12}$$

If the area and the scale of the network is stable, the P is related to the variable λ. We have simulation experiments to figure out how different variables of m, n, and λ affect the secure-connectivity P in RD-KDS. Assume there are m=100 gateways in the network. We have 4 situations: (1) n=2000; (2) n=4000; (3) n=6000; (4) n=8000. That is, the average size of a cluster is: (1) t_a=20; (2) t_a=40; (3) t_a=60; (4) t_a=80. In Fig. 8, the x-axis is the number of parent-gateways λ, and the y-axis is the secure-connectivity P. The experiment results show that, in a network with given m and n, the secure-connectivity P is increased when the λ increases. This coincides exactly with the Eq. (11).

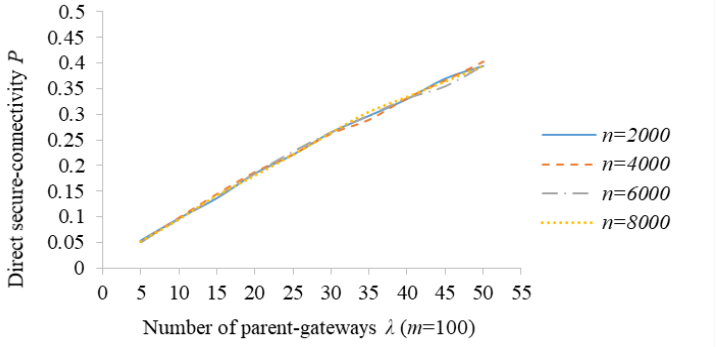


Figure 8: Direct secure-connectivity P *vs.* number of parent-gateways λ in RD-KDS

We can further improve the probability of direct key establishment according to training the Group Deployment Model in GD-KDS.

5.3 Overheads

Denote the overheads of a gateway in Direct and Indirect Key Distribution steps as *Di_Ov* and *In_Ov*. So we have the average overhead of gateway to distribute a key to a sensor is *Av_Ov*in Eq. (13).

$$Av\_Ov = P \cdot Di\_Ov + (1 - P) \cdot In\_Ov \quad . \tag{13}$$

Since the average amount of sensors located in the coverage area of a gateway is n_a, hence, we have the total average overhead of gateway in both direct and indirect key distribution steps is achieved in Eq. (14).

$$Total\_Av\_Ov = n_a \cdot Av_{Ov} = n_a \cdot \left[\left(1 - \frac{C_{n-t_a}^{n_a}}{C_n^{n_a}} \right) \cdot Di\_Ov + \frac{C_{n-t_a}^{n_a}}{C_n^{n_a}} \cdot In\_Ov \right] \quad . \tag{14}$$

In this paper, we mainly consider the overhead in terms of storage, communication and computation. The communication overhead is measured by the transmission rounds and the length of the packages. The computation overhead is mainly focused on PUF operation and encryption or decryption operations, e.g., AES. The storage overhead in the sensor can be omitted because the PUF is realized by hardware and the sensor is pre-distributed with no key in *Initialization* step. The storage overhead in the gateway is characterized by the memory size of tuples that are pre-loaded in *Initialization* step.

Tab. 2 summaries the gateway and sensor comprehensive overheads in both Direct and Indirect Key Distribution steps of this RD-KDS. We denote key length as α. The PUF CRP length (including C and R) is β, and we use ε as an adjust ratio to equalize the CRP length and key length: $\alpha=\varepsilon\beta$, where ε is related to the PUF physical structure. Node *id* length is denoted as ω. Symmetric encryption, e.g., AES, is used in the algorithm procedure, so the length of ciphertext is equal to the plaintext. PUF is realized by hardware, while AES can be realized by hardware or software. We denote the calculated overhead of PUF and AES algorithms as C_P and C_{ED}, respectively.

Table 2: Overhead in key distribution step

Direct Key Distribution *P*	Communication	Gateway	2α
		Sensor	α
	Computation	Gateway	$2\ C_{ED}$
		Sensor	$C_P+2\ C_{ED}$
	Storage	Gateway	$\omega+2\alpha$
		Sensor	0
Indirect Key Distribution 1-*P*	Communication	Gateway	$\omega+3\alpha$
		Gateway	3α
		Sensor	α
	Computation	Gateway	C_{ED}
		Gateway	$3\ C_{ED}$
		Sensor	$C_P+2\ C_{ED}$
	Storage	Gateway	0
		Gateway	$\omega+2\alpha$
		Sensor	0

6 Grouping deployment key distribution scheme (GD-KDS)

In this paper, we define the secure-connectivity as the probability that a gateway can establish a pairwise key with a sensor in its coverage area. In the proposed Random Deployment Scheme in *Section 5*, a gateway distributes a key to a sensor by direct or indirect ways. In this section, we propose a GD-KDS to further improve the probability of direct key distribution by training a Grouping Deployment Model.

6.1 Proposal

6.1.1 Step 1: grouping model

The grouping deployment model [Liu, Ning and Du (2008)] is employed. Before deployment, all the nodes (including sensors and gateways) are divided into m Deployment Groups (DG), denoted as $\{DG_i\}_{i=0,\ldots,m-1}$. In each deployment group, there is 1 gateway and $d=n/m$ sensors, where we call the gateway as the parent-gateway of these d sensors. Fig. 9 gives an example of the grouping model, in which, the number of gateways is m=4 and the number of sensors is n=12. In this model, all nodes are divided into 4 DGs, and in each DG there is 1 gateway and 3 sensors. Nodes in one DG will be thrown into the destination area together, so as to increase the probability of forming a cluster and distributing keys directly.

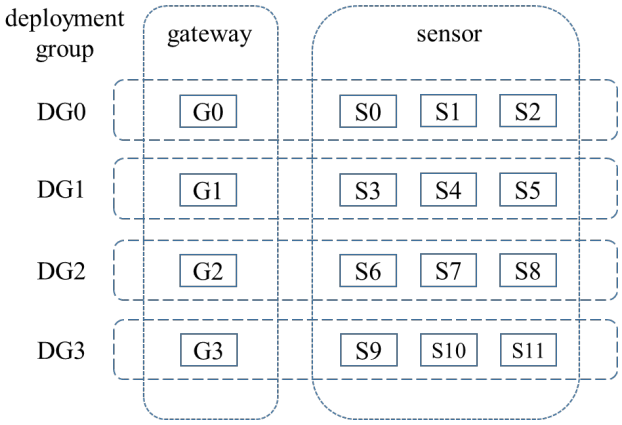


Figure 9: An example of the grouping model, in which, m=4 and n=12

6.1.2 Step 2: initialization

Before network deployment, each sensor S is embedded with a PUF, denoted as Γ_S. Take a random challenge number C_S as the input of Γ_S and get the output response R_S. In a deployment group DG_i, the gateway is named as the parent-gateway of all sensors in DG_i, and save a CRP tuple $<id_S, C_S, R_S>$ of each sensor into the gateway.

For example, in Fig. 9, in DG_0, take the gateway G_0 as the parent-gateway of sensor S_0, S_1, and S_2. Generate a CRP tuple for each sensor as $<id_{S0}, C_{S0}, R_{S0}>$, $<id_{S1}, C_{S1}, R_{S1}>$, $<id_{S2}, C_{S2}, R_{S2}>$, and save them into G_0.

6.1.3 Step 3: grouping deployment

The destination area is divided into $g \times g$ zones. A group of nodes (including a gateway and n/m sensors) are deployed together and assumed to be located in one zone. Therefore, there are $(1+n/m)/g^2$ nodes in each zone. Different with the RD-KDS, nodes are deployed by groups and each group is thrown into the same zone at the same time. After network deployment, the gateway launches the cluster forming process. Nodes in the same deployment group form a cluster with high probability since they are close to each other.

A simple example is shown in Fig. 10, in which the destination area is divided into 4 zones. Nodes are grouping into 4 deployment groups, and each group DG_i locate in a $Zone_i$ to form a $Cluster_i$. It shows an ideal deployment example here because it is 100% consistent with the deployment model in Fig. 9.

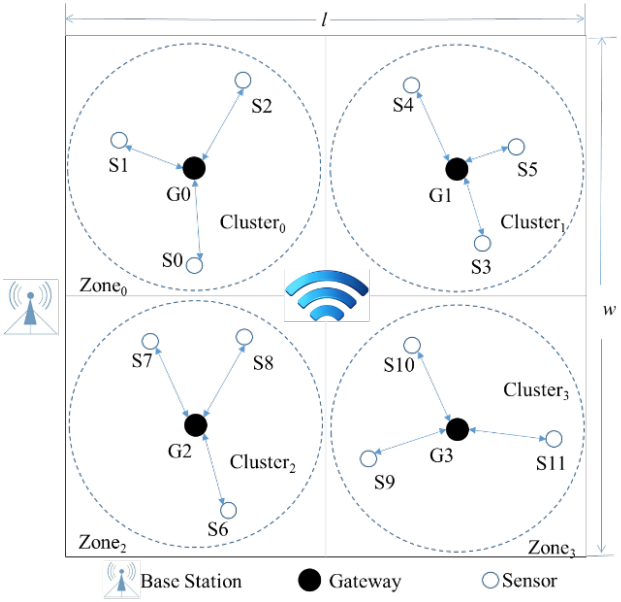


Figure 10: The network deployment result is ideally consistent with the deployment model in Fig. 9

6.1.4 Step 4: key distribution

In each cluster $Cluster_i$, the gateway G_i can directly distribute a session key to a sensor S_j if they are belonging to the same deployment group DG_i. In Fig. 10, since it is loaded with the CRP tuple $<id_{S0}, C_{S0}, R_{S0}>$ of sensor S_0 in the *Initialization* step, the gateway G_0 is able to directly authenticate and distribute a session key to S_0 via the Basic-KDS.

6.1.5 Step 5: indirect key distribution

However, the ideal result is very difficult to be achieved in real deployment environment. Some sensors mightily are not located into their parent gateways' coverage area. In this situation, an Indirect Key Distribution will be processed by the similar way in RD-KDS.

6.2 Secure-connectivity

Suppose the number of gateways and sensors is m and n. And suppose the network area is a rectangle with the length is l and the width is w. The coverage area of a gateway is a circle with the radius is r. The destination area is divided into g^2 zones.

The average amount of sensors located in the coverage area of a gateway is n_a in Eq. (15).

$$n_a = \frac{n\pi r^2}{(l \times w)/g^2} = \frac{n\pi r^2 g^2}{l \times w} \quad . \tag{15}$$

The amount of tuples that are saved in a gateway is t_a in Eq. (16).

$$t_a = \frac{n}{m} \quad . \tag{16}$$

So, the fraction that the gateway is loaded with a sensor's tuple is denoted as p_t in Eq. (17).

$$p_t = 1 - \frac{C_{n-t_a}^{n_a}}{C_n^{n_a}} \quad . \tag{17}$$

And the secure-connectivity that a gateway can directly authenticate and distribute key with a sensor is P in Eq. (18).

$$P = p_t \quad . \tag{18}$$

In a given network with stable m and n, the secure-connectivity P can be raised by increasing the deployment zones. This is proved by Fig. 11. We have 4 situations: (1) m=100, n=10000; (2) m=100, n=6000; (3) m=200, n=10000; (4) m=200, n=6000. In Fig. 11, the x-axis is g, and the y-axis is P. This coincides exactly with the Eq. (17).

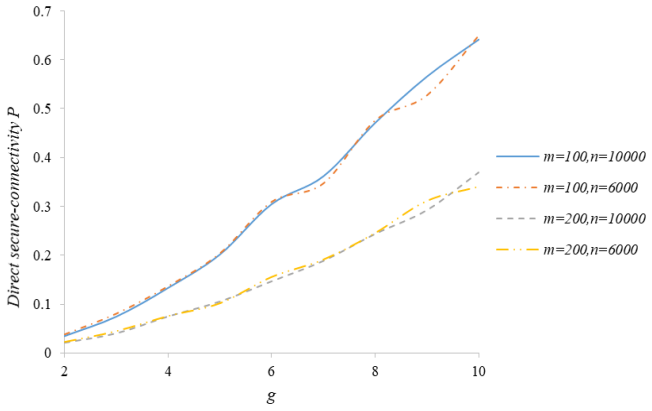


Figure 11: Direct secure-connectivity P *vs.* g in GD-KDS

6.3 Overheads

In this GD-KDS, we have the total average overhead of gateway in both direct and indirect key distribution steps is achieved in Eq. (19).

$$Total_{Av_{Ov}} = \frac{n}{m} \cdot Av_{Ov} = \frac{n}{m} \cdot [P \cdot Di_{Ov} + (1 - P) \cdot In_{Ov}] \tag{19}$$

Di_Ov and *In_Ov* have the same definition and measurement as Tab. 2.

7 Experiments and comparison

In this section, we make some experiments to compare the RD-KDS and GD-KDS on secure-connectivity. We compare the secure-connectivity in RD-KDS and GD-KDS when the gateway has the same storage overheads. In both schemes, the gateway is loaded with 1 PUF CRP tuple for 1 sensor. In Fig. 12, the x-axis is the number of PUF CRPs saved in a gateway, and y-axis is the secure-connectivity P. Suppose there m=100 gateways. It is shown in Fig. 12, the GD-KDS has much improved secure-connectivity than RD-KDS by employing the Grouping Deployment strategy.

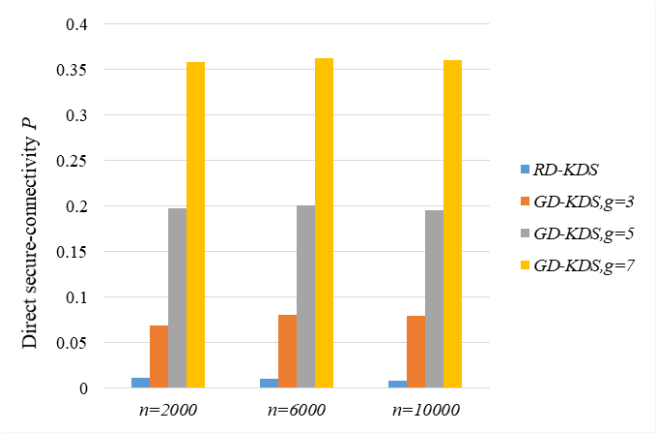


Figure 12: Direct secure-connectivity P *vs.* g in GD-KDS

8 Conclusion

In this paper, we utilized the physical unclonable function (PUF) to design a Basic Key Distribution Scheme (Basic-KDS) in wireless sensor networks. Then we proposed a Random Deployment Key Distribution Scheme (RD-KDS) and a Grouping Deployment Key Distribution Scheme (GD-KDS) by combining the Basic-KDS with random deployment model and group deployment model, respectively. When similar secure-connectivity was achieved in these two proposals, RD-KDS required the gateway to save more PUF challenge-response pairs (CRPs) than GD-KDS. GD-KDS could substantially improve the secure-connectivity in the premise of proper grouping and positioning. By applying a PUF as a key management and authentication module, the sensor was pre-distributed with 0 keys in memory. It did not only save the storage overhead, but also provided perfect resilience against sensor capture attacks. Besides, the two-way authentication between a gateway and a sensor was also guaranteed based on the PUF CRPs. Proof and experiments were given to analyze the security and performances.

Funding Statement: This work is supported by the National Natural Science Foundation of China (under grant 61902163), the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (under grant 17KJD520003, 19KJB520033), and the Research Startup Foundation of Jinling Institute of Technology (under grant JIT-B-201639, JIT-B-201726, JIT-B-202001).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Alagheband, M.; Aref, M. (2012): Dynamic and secure key management model for hierarchical heterogeneous sensor networks. *IET Information Security*, vol. 6, no. 4, pp. 271-280.

Aman, M. N.; Chua, K. C.; Sikdar, B. (2016): Position paper: physical unclonable functions for IoT Security. *Proceedings of 2nd ACM International Workshop on IoT Privacy, Trust, and Security*, pp. 10-13.

Bahrampour, R.; Atani, R. (2013): A novel key management protocol for wireless sensor networks based on PUFs. *International Journal of Future Generation Communication and Networking*, vol. 6, no. 2, pp. 93.

Blom, R. (1985): An optimal class of symmetric key generation system. *Proceedings of the EUROCRYPT 84 workshop on Advances in Cryptology: Theory and Application of Cryptographic Techniques*, pp. 445-448.

Bohge, M.; Trappe, W. (2003): An authentication framework for hierarchical ad hoc sensor networks. *Proceedings of the Second ACM Workshop on Wireless Security*, pp. 79-87.

Boujelben, M.; Cheikhrouhou, O.; Abid, M.; Youssef, H. (2009): Establishing pairwise keys in heterogeneous two-tiered wireless sensor networks. *Proceedings of 3rd International Conference on Sensor Technologies and Applications Athens*, pp. 18-23.

Carman, D.; Kruus, P.; Matt, B. (2000): Constraints and approaches for distributed sensor network security (final). *NAI Labs Technical Report*, pp. 1-139.

Chen, C. Y.; Chao, H. C. (2014): A survey of key distribution in wireless sensor networks. *Security and Communication Networks*, vol. 7, no. 12.

Du, X. J.; Xiao, Y.; Guizani, M.; Chen, H. H. (2007): An effective key management scheme for heterogeneous sensor networks. *Ad Hoc Networks*, vol. 5, pp. 24-34.

Eschenauer, L.; Gligor, V. D. (2002): A key management scheme for distributed sensor networks. *Proceedings of 9th ACM Conference on Computer and Communication Security*, pp. 41-47.

Fang, L. M.; Li, Y.; Yun, X. Y.; Wen, Z. Y.; Ji, S. L. et al. (2019): THP: a novel authentication scheme to prevent multiple attacks in SDN-based IoT network. *IEEE Internet of Things Journal*.

Fukushima, K.; Hidano, S.; Kiyomoto, S. (2016): Sensor-based wearable PUF. *International Conference on Security and Cryptography*, pp. 207-214.

Gassend, B.; Clarke, D.; Dijk, M.; Devadas, S. (2002): Silicon physical random functions. *Proceedings of the Computer and Communication Security Conference*.

Ge, C. P.; Liu, Z.; Xia, J.; Fang, L. M. (2019): Revocable identity-based broadcast proxy re-encryption for data sharing in clouds. *IEEE Transactions on Dependable and Secure Computing*.

Guajardo, J.; Kumar, S.; Tuyls, p. (2008): Key distribution for wireless sensor networks and physical unclonable functions. *Secure Component and System Identification*, pp. 17-18.

Huang, M.; Yu, B.; Li, S. (2018): PUF-assisted group key distribution scheme for software-defined wireless sensor networks. *IEEE Communications Letters*, vol. 22, no. 2, pp. 404-407.

Jolly, G.; Kuscu, M. C.; Kokate, P.; Younis, M. (2003): A low-energy key management protocol for wireless sensor networks. *Proceedings of 8th IEEE International Symposium on Computers and Communication*, pp. 335-340.

Kadri, B.; Feham, M.; Mhammed, A. (2012): Architecture aware key management scheme for wireless sensor networks. *International Journal of Information Technology & Computer Science*, vol. 4, no. 12, pp. 50-59.

Kumar, D. A.; Sherali, Z.; Debiao, H. (2018): Taxonomy and analysis of security protocols for Internet of Things. *Future Generation Computer Systems*, vol. 89, pp. 110-125.

Lee, J. W.; Lim, D.; Gassend, B.; Suh, G. E.; Marten, V. D. (2004). A technique to build a secret key in integrated circuits for identification and authentication applications. *Proceedings of IEEE VLSI Circuits Symposium*, pp. 176-179.

Li, G. S.; Yan, J. H.; Chen, L.; Wu, J. H.; Lin, Q. Y. et al. (2019): Energy consumption optimization with a delay threshold in cloud-fog cooperation computing. *IEEE Access*, vol. 7, no. 1.

Liu, D.; Ning, P.; Du, W. (2008): Group-based key predistribution for wireless sensor networks. *ACM Transactions on Sensor Networks*, vol. 4, no. 2, pp. 1-30.

Mao, Y.; Zhang, J.; Qi, H.; Wang, L. (2019): DNN-MVL: DNN-multi-view-learning-based recover block missing data in a dam safety monitoring system. *Sensors*.

Mukhopadhyay, D. (2016): PUFs as promising tools for security in internet of things. *IEEE Design & Test*, vol. 33, no. 3, pp. 103-115.

Pappu, R.; Recht, B.; Taylor, J; Gershenfeld, N. (2002): Physical one-way functions. *Science*, vol. 297, no. 5589, pp. 2026-2030.

Rahman, M. T.; Rahman, F.; Forte, D; Tehranipoor, M. (2016): An aging-resistant RO-PUF for reliable key generation. *IEEE Transactions on Emerging Topics in Computing*, vol. 99, no. 3, pp. 335-348.

Ren, Y. J.; Zhu, F. J.; Sharma, P. K.; Wang, T.; Wang, J. et al. (2020): Data query mechanism based on hash computing power of blockchain in internet of things. *Sensors*, vol. 20, no. 1.

Simplício, M. A.; Barreto, P.; Margi, C. B.; Carvalho, T. (2010): A survey on key management mechanisms for distributed wireless sensor networks. *Computer Networks*, vol. 54, no. 15, pp. 2591-2612.

Tuyls, P.; Schrijen, G. J.; Skoric, B.; Geloven, J. V.; Wolters, R. (2006): Read-proof hardware from protective coatings. *Proceedings of 8th International Workshop of Cryptographic Hardware and Embedded Systems*, pp. 363-383.

Wang, C. X.; Shao, X.; Gao, Z.; Zhao, C. X.; Gao, J. (2019): Common network coding condition and traffic matching supported network coding aware routing for wireless

multihop network. *International Journal of Distributed Sensor Networks*, vol. 15, no. 6, pp. 1-20.

Younis, M.; Youssef, M.; Arisha, K. (2002): Energy-aware routing in cluster-based sensor networks. *Proceedings of the 10th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 129-136.

Zhang, J. Q.; Varadharajan, V. (2010): Wireless sensor network key management survey and taxonomy. *Journal of Network and Computer Applications*, vol. 33, no. 2, pp. 63-75.

Zhang, S.; Chang, Y.; Yan, L. L.; Sheng, Z. W.; Yang, F. et al. (2019): Quantum communication networks and trust management: a survey. *Computers, Materials & Continua*, vol. 61, no. 3, pp. 1145-1174.