Hindawi
Security and Communication Networks
Volume 2021, Article ID 5532683, 13 pages
https://doi.org/10.1155/2021/5532683

Research Article

PUF-Based Mutual-Authenticated Key Distribution for Dynamic Sensor Networks

Yanan Liu,[1] **Yijun Cui**,[2] **Lein Harn**,[3] **Zheng Zhang**,[1] **Hao Yan**,[1] **Yuan Cheng**,[1] **and Shuo Qiu**[1]

[1]*School of Network Security, Jinling Institute of Technology, Nanjing 211169, China*
[2]*College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 211100, China*
[3]*Department of Computer Science Electrical Engineering, University of Missouri, Kansas City 64110, MO, USA*

Correspondence should be addressed to Zheng Zhang; zhangzheng@jit.edu.cn

Received 5 February 2021; Accepted 22 April 2021; Published 3 May 2021

Academic Editor: Qing Yang

Because of the movements of sensor nodes and unknown mobility pattern, how to ensure two communicating (static or mobile) nodes authenticate and share a pairwise key is important. In this paper, we propose a mutual-authenticated key distribution scheme based on physical unclonable functions (PUFs) for dynamic sensor networks. Compared with traditional key predistribution schemes, the proposal reduces the storage overhead and the key exposure risks and thereby improves the resilience against node capture attacks. Mutual authentication is provided by the PUF challenge-response mechanism. However, the PUF response is not transmitted in plain forms so as to resist the modelling attacks, which is vulnerable in some existing PUF-based schemes. We demonstrate the proposed scheme to improve the secure connectivity and other performances by analysis and experiments.

1. Introduction

Many applications of wireless sensor networks (WSNs) are working in hostile battlefield environments or unmanned areas with poor conditions. Sensor nodes and wireless channels are vulnerable to malicious attacks, such as physical capture nodes, data tampering, and side channel attacks [1–3]. Data encryption is a crucial technology to ensure secure communication between the cloud and end-devices [4–6]. The authentication and key distribution are the premise and foundation [7, 8].

In 2002, Eschenauer and Gligor proposed a random key predistribution scheme [9] for the resource limited sensors. In 2007, Du et al. applied Eschenauer's scheme into hierarchical sensor networks and proposed an asymmetric key predistribution scheme (AP) [10]. This kind of "probabilistic" schemes had low computation and communication overhead but cannot ensure that any two of communicating nodes share a pairwise key. Besides, the key storage amount showed a tradeoff between the network connectivity and resilience against node capture attacks. In 2009, Boujelben proposed a key management scheme based on the Blom matrix [11] to improve the resilience against node capture; however, the computation cost for matrix operation was too complicated for common sensors [12]. In terms of public key algorithms, in 2012, Benamar et al. [13] proposed a dynamic security key management model for hierarchical sensor networks based on public key infrastructure (PKI). In 2015, Lee and Kim [14] proposed a key renewal scheme with sensor authentication under clustered wireless sensor networks based on modular exponentiation which was similar to the Diffie–Hellman key exchange. These schemes increased the connectivity; however, the public key computational overhead was too large for sensors. In 2010, Han et al. [15] proposed an approach for dynamic node authentication and key exchange, which reduces the overhead of mobile node reauthentication. Each sink node authenticates other neighboring sink and sensor nodes and supports reauthentication with less communication and computation overhead. In 2015, Erfani et al. [16] proposed a

key management scheme, which used key predistribution and postdeployment key establishment mechanisms for dynamic sensor networks. The predistributed keys are loaded to the memory of sensor nodes before network deployment, and after that, some postdeployment keys are generated and stored in each sensor node. In Erfani's approach, the base station is involved in intracluster authentication and key distribution, which costs too much communication overheads. In 2020, Tian et al. [17] proposed a blockchain-based secure key management scheme with trustworthiness in dynamic wireless sensor networks, which designed a secure cluster formation algorithm and a secure node movement algorithm to implement key management.

This paper proposed a mutual authenticated key distribution scheme based on physical unclonable functions (PUFs) in dynamic sensor networks, so as to help the sink node to authenticate and distribute session keys to the static and mobile sensors. Lightweight mutual authentication is guaranteed by a challenge-response mechanism based on the PUF. To address the PUF challenge-response pairs (CRPs) exposure problem, the CRPs are not transmitted as plaintext in order to resist the modelling attack to PUF. In addition, sensors are not required to prestore any keys in memory, which not only saves the storage overhead but also improves the resilience against sensor node capture attacks.

2. Physical Unclonable Function (PUF)

2.1. Review of PUFs. Physical unclonable function (PUF) is a new encryption component that can extract random differences introduced by inconsistencies in manufacturing processes between gate circuits or connection lines (wires) in integrated circuits (IC). These random differences can be used to generate an encrypted (response) signal with certain rules [18]. Random differences in a physical object can be interpreted as the unique "fingerprint" of a hardware instant. In addition to IC PUFs [19], there are silicon PUFs [20], coated PUFs [21], and so on. We use a one-way mapping function P to describe PUF, which can be expressed as

$$P\colon C \longrightarrow R\colon P(c) = r, \quad c \in C, r \in R. \tag{1}$$

The functional mapping between input c and output r is instance-specific and unpredictable prior to the actual fabrication of the circuit. When an electrical stimulus is applied to the structure, it reacts in an unpredictable (but instance-wise repeatable) manner due to the complex interaction of the stimulus with the physical microstructure of the device. The exact nature of this microstructure depends on physical factors introduced during manufacturing. The applied stimulus is considered as the "challenge," while the reaction generated by the PUF is considered as the "response." A specific challenge and its response together form a challenge-response pair (CRP) (c, r), and the CRP dataset acts as a unique fingerprint for the instance.

The attractive features of PUFs are light-weightiness, unpredictability, unclonability, and uniqueness, which make PUFs valuable in designing ultralightweight authentication, key generation, and other security protocols [22, 23]. Device authentication is the process that an authenticator verifies the identity of a device client before communication. PUF CRP can be implemented in the challenge-response authentication mechanism. The authenticator creates a CRP database that stores all the challenges and their expected responses from registered clients. To verify the identity of a client, the authenticator first selects a challenge from the database and sends it to the client. The client generates a response to the challenge using its on-board PUF and provides it to the authenticator. By comparing the current client's response against the one stored in the CRP database, the authenticator infers whether the client is trusted or not.

This new type of schemes speeds up the authentication process and also lightens the key storage and thereby reduces key exposure risk. A PUF with a large enough challenge space to make exhaustive enumeration of its CRP set infeasible is termed a strong PUF and is the PUFs of choice in most practical security applications. We keep ourselves confined to strong PUFs in this work. Since the assessment of a PUF implies a physical measurement, it is very susceptible to circuit noise. Hence, to make it reliable and to have full entropy, [22] had proposed an error correction circuit with a very low hardware overhead to reduce the fuzziness of the PUF's responses and make it more robust and reliable. However, in our work, we consider each PUF structure 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.

In 2015, Allam proposed a scheme that depends on the physical layer mechanisms, which consist of PUF and Channel Status Information (CSI) for providing point-to-point real-time hardware-based authentication technique between two parties communicating directly through wireless media and effective key exchange to assure an authenticated secure channel between them [23]. In 2013, Bahrampour and Atani proposed a Key Management Protocol for Wireless Sensor Networks based on PUFs, in which the PUFs were used to design the public keys [24]. In 2017, Chatterjee et al. proposed a PUF-based secure communication protocol for PUF [25]. The PUF was used to generate the public key based on the bilinear pairing of each device in the key agreement protocol. In 2018, Braeken improved Chatterjee's protocol efficiency by way of employing the Elliptic Curve Qu Vanstone (ECQV) [26]. In 2019, Li et al. proposed a PUF-based secure communication system for the Internet of Things [27]. In 2020, Zhang et al. proposed a PUF-based Key Distribution in Wireless Sensor Networks [28].

2.2. Configurable RO PUFs. The PUF circuit, which is the core of authentication and key distribution in our scheme, should be easily implemented on the FPGA with good uniqueness and reliability. In our previous work, several types of configurable RO PUF are proposed, including MUX based RRO PUF in [29], XOR gate based XCRO PUF in [30], and tristate configurable TCRO PUF in [31]. In this paper, the MUX based RRO PUF is chosen. The MUX based configurable RO (CRO) PUF was first introduced in [32],

where each ring oscillator can be reconfigured by using a multiplexer to select one of two inverters that are connected to the multiplexer to form an RO. Our reconfigurable RRO design, as shown in Figure 1, is consisted of a chain of inverter delay units and an AND gate delay unit. When the configurable signal of a MUX is "0," the upper path will be chosen. On the contrary, when the signal is "1," the lower path will be chosen to construct the RO structure. The configure procedure extracts the transfer difference of each MUX and the delay of the upper and lower path.

2.3. Implement of PUFs. The PUF used in our approach is implemented and studied based on Xilinx SoC FPGAs and will be applied to real-world scenarios based on ASIC or SoC FPGA including ARM core (e.g., Xilinx Zynq-7000 series, Altera SoC or Microsemi Smart Fusion2) after validation. As shown in Figure 2, the main components include MUX, XOR gate, inverters, and AND gate. In the implementation of the RRO PUF, the primitive MUXF7 is chosen for the multiplexer, the primitive LUT1 is adopted for the inverter, and LUT2 is utilized for the AND gate. Eight delay units that include seven inverter delay units and one AND gate delay unit are included in the single RRO array. Each delay unit occupies one slice and two delay units can be implemented in one configurable logic block (CLB). Therefore, four CLBs are needed to implement one RRO PUF array. In order to make sure that all RROs are identically routed, they are created as hard macros to avoid the bias introduced in the placement and routing. The detailed design can be referred in authors' previous work [29].

3. PUF-Based Mutual Authentication and Key Distribution

3.1. Network Model. Large-scale wireless sensor networks are usually deployed in a hierarchical clustered structure and contain heterogeneous nodes, such as a base station (BS), several sink nodes (SN), and a number of low-energy sensors. BS is assumed to be resourceful and global trusted. It manages the entire network and stores all gathered information by sensor nodes. Sink node is assumed to have higher hardware configurations than sensors, including memory, communication, and computation ability. A sink node acts as a gateway between sensor nodes and BS. Sensors are divided into nonoverlapping clusters; they collect data from surroundings and send raw data to the sink node. Sensor nodes are assumed to have a random linear movement pattern, while the BS and sink nodes are static like Han and Erfani schemes [15, 16]. Because of unpredictable position of mobile sensors, how to ensure a sink node to authenticate and distribute a pairwise key to every present cluster-member sensor is difficult.

In our network model, assume there are n sensors, named $S_{0,\ldots,n-1}$, and m sink nodes, named $\mathrm{SN}_{0,\ldots,m-1}$. Each sensor node has a unique ID S_i and embeds a chip with a PUF structure, denoted as P_{Si}. Before network deployment, all the nodes are divided into m deployment groups (DGs), denoted as $\{\mathrm{DG}_i\}_{i=0,\ldots,m-1}$. In each DG, there is 1 SN and

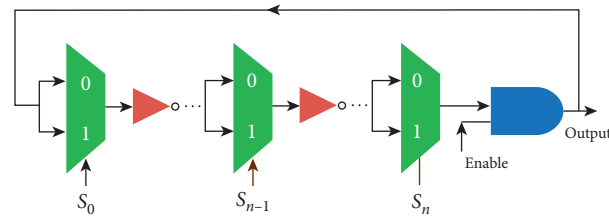

Figure 1: RRO PUF structure.

$d = n/m$ sensors, and the SN is called the "Home-SN" of these d sensors. Nodes in a DG will be thrown into the destination area together, so as to form a cluster. Figure 3 gives an example with 3 DGs and 9 sensors.

3.2. Initialization and Network Deployment. Before network deployment, for each sensor S_i, take a random challenge number c_{Si} as the input of PUF P_{Si} and get the output response r_{Si}; prestore the PUF CRP (c_{Si}, r_{Si}) to the Home-SN of S_i by indexing with the sensor ID S_i. For example, in Figure 3, in DG0, take the sink node SN0 as the Home-SN of sensors S_0, S_1, and S_2. Generate a CRP for each sensor as (c_{S0}, r_{S0}), (c_{S1}, r_{S1}), and (c_{S2}, r_{S2}) and save them into the memory of SN0.

After network deployment, the sink node launches the cluster forming process (not discussed in this paper, please refer to [33]), which divides all sensor nodes into clusters with no cross coverage. Each cluster includes a sink node, which is called the "cluster head" (CH), and n/m sensors, which are called the "cluster members" (CM). Nodes in the same DG form a cluster with very high probability since they are thrown close to each other. It shows an ideal deployment example in Figure 4.

In order to ensure the secure intracluster communication, a sink node needs to authenticate and distributes a pairwise key to every cluster-member sensor. In a short period after network deployment, assume sensors are static. It is easy for the sink node to run the authentication and key distribution according to the challenge-response mechanism based on PUF CRP. However, after some working time, a sensor moves into another cluster's region (as shown in Figure 5), in which the sink node does not share the PUF CRP of the mobile sensor. In this situation, the sink node in the present cluster, called the "Present-SN," should authenticate the mobile sensor via the help of the Home-SN. In the following section, we will describe our approach by two subschemes for static sensors and mobile sensors, respectively.

The differences between these two subschemes mainly happened in the following aspects: (1) there were two entities in static subscheme: Home-SN and the sensor; there were three entities in mobile subscheme: Home-SN, Present-SN, and the sensor; (2) in the static subscheme, the (Present also Home) SN generated the session key with the sensor; in the mobile subscheme, the Home-SN generated the session key between the Present-SN and the sensor; (3) in the static subscheme, the (Present also Home) SN authenticated the sensor directly; in the mobile subscheme, the Home SN-helped the Present-SN to authenticate the sensor.

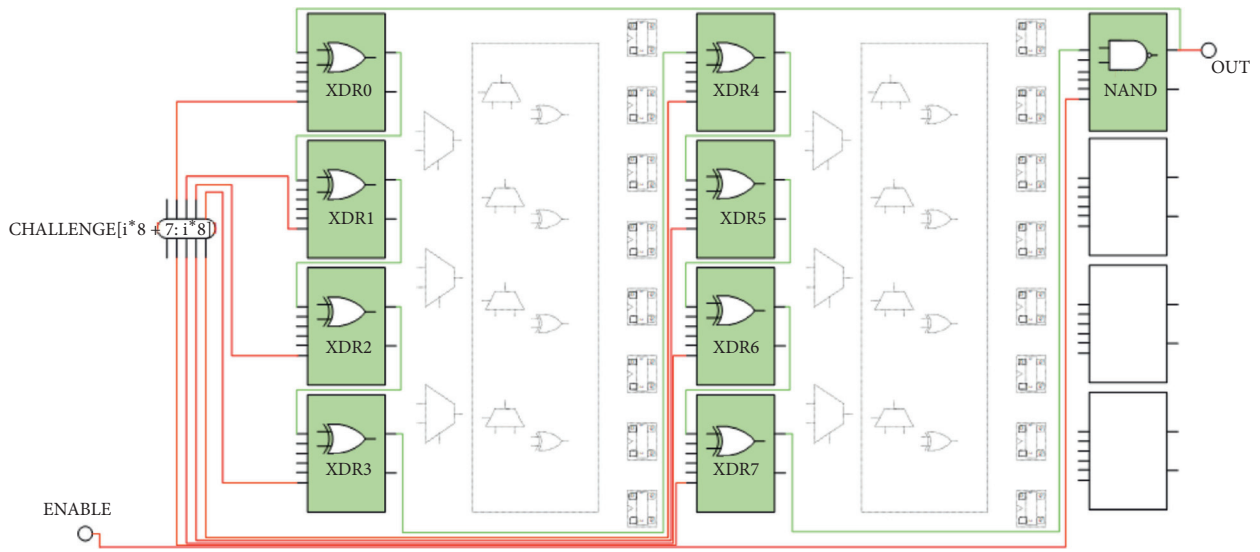

Figure 2: Implementation of an RRO in a CLB.

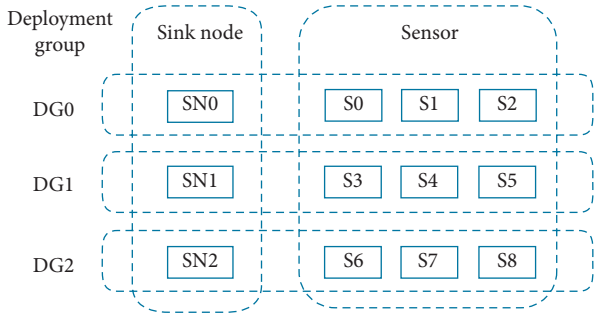

Figure 3: An example of the deployment model with 3 sink nodes and 9 sensors.

3.3. Static Sensors Subscheme. The approach of a sink node SN_0 authenticating and distributing a pairwise key to a static sensor S_0 is described as shown in Figure 6.

(1) After network deployment and clustering process, in the cluster C_0, the sink node SN_0 detects a sensor S_0 in its cluster. SN_0 reads a PUF CRP in its memory: (c_{S0}, r_{S0}) by indexing of id_{S0}.

(2) SN_0 computes a temporary key key_{SN0}:

$$key_{SN0} = H\left(\|r_{S0}\text{timestamp1}\right), \tag{2}$$

where H is a hash function.

SN_0 generates a session key key_{SN0-S0} and encrypts it by key_{SN0} to get cipher1:

$$\text{cipher1} = E\left(key_{SN0}, key_{SN0-S0}\right). \tag{3}$$

E is symmetric encryption (e.g., AES). Then, SN_0 encrypts c_{S0} by using key_{SN0-S0}:

$$\text{cipher2} = E\left(key_{SN0-S0}, \|c_{S0}\text{timestamp1}\right). \tag{4}$$

Then, SN_0 generates a secret random number nonce1 and encrypts it by using key_{SN0-S0}:

$$\text{cipher3} = E\left(key_{SN0-S0}, \text{nonce1}\right). \tag{5}$$

SN_0 sends the challenge c_{S0}, cipher1, cipher2, and cipher3 to S_0:

$$SN_0 \longrightarrow S_0: \|c_{S0}\|\text{cipher1}\|\text{cipher2}\|\text{cipher3timestamp1}. \tag{6}$$

(3) After receiving the message, the sensor S_0 firstly inputs c_{S0} into the PUF structure P_{S0}, which is embedded during the initialization phase, and gets the output response r_0:

$$r_0 = P_{S0}\left(c_{S0}\right). \tag{7}$$

S_0 computes a temporary key, key_{S0}:

$$key_{S0} = H\left(\|r_0\text{timestamp1}\right). \tag{8}$$

Then, S_0 decrypts the cipher1 to get the pairwise key, key_{S0-SN0}:

$$\begin{aligned}\text{plain1} &= D\left(key_{S0}, \text{cipher1}\right)\\ &= D\left(key_{S0}, E\left(key_{SN0}, key_{SN0-S0}\right)\right) = key_{S0-SN0}.\end{aligned} \tag{9}$$

The function D is the decryption operation of E.

S_0 decrypts cipher2 by using key_{S0-SN0} and gets plain2:

$$\begin{aligned}\text{plain2} &= D\left(key_{S0-SN0}, \text{cipher2}\right)\\ &= D\left(key_{S0-SN0}, E\left(key_{SN0-S0}, \|c_{S0}\text{timestamp1}\right)\right).\end{aligned} \tag{10}$$

The sensor S_0 checks if the equation plain2 = $\{c_{S0}$ ‖timestamp1$\}$ is correct.

If not, S_0 deduces that the sink node SN_0 is not its valid Home-SN, since it does not share a correct PUF

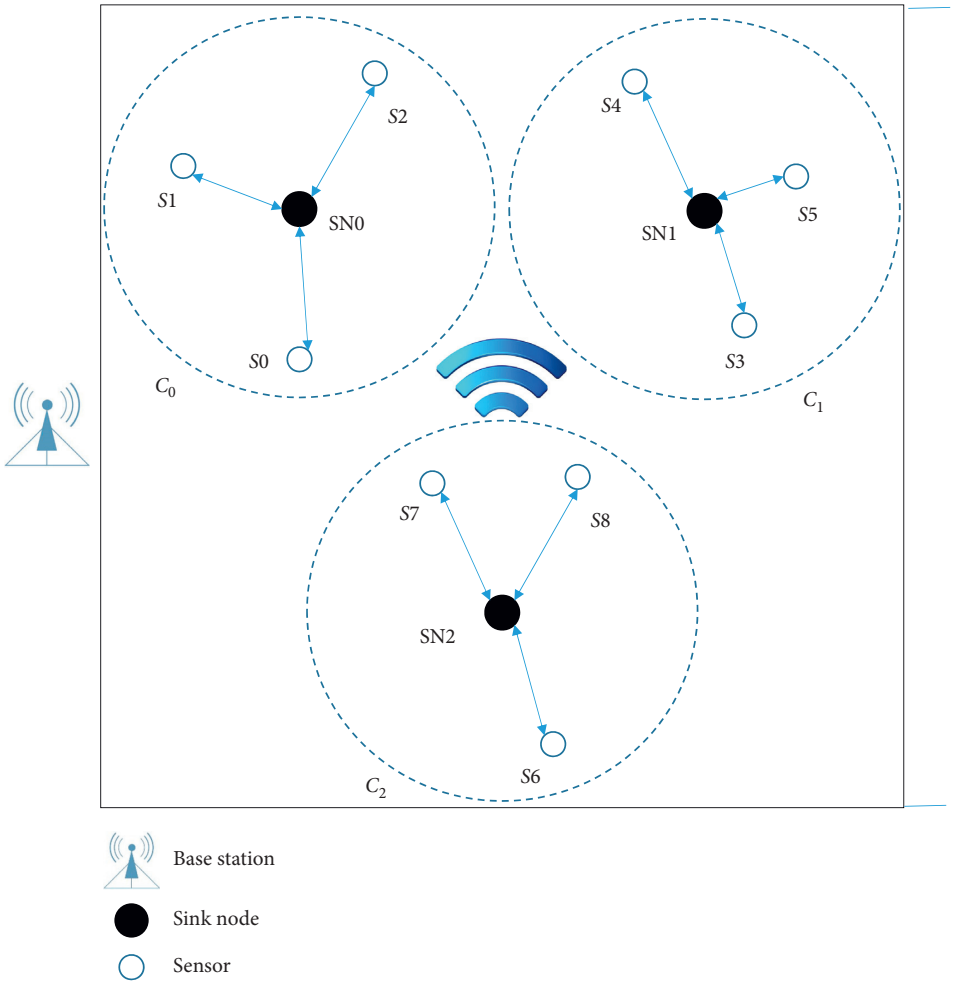

Figure 4: The network deployment.

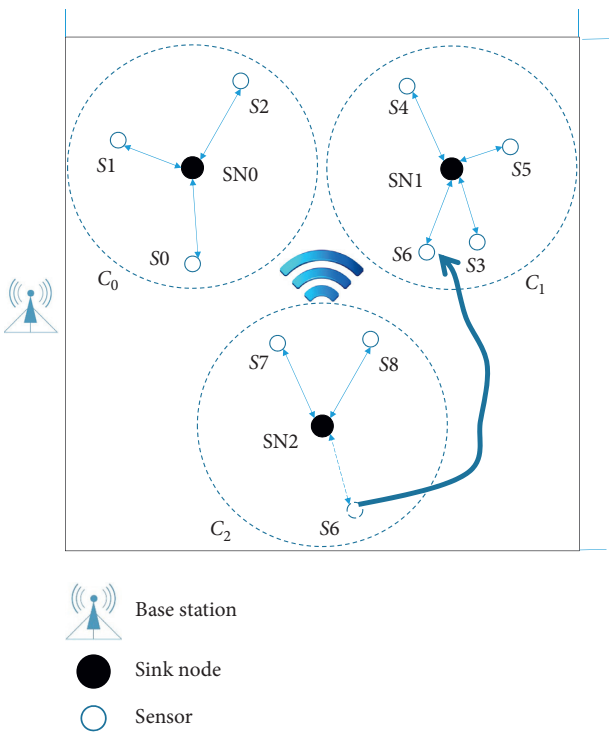

Figure 5: S_6 moves from C_2 into C_1.

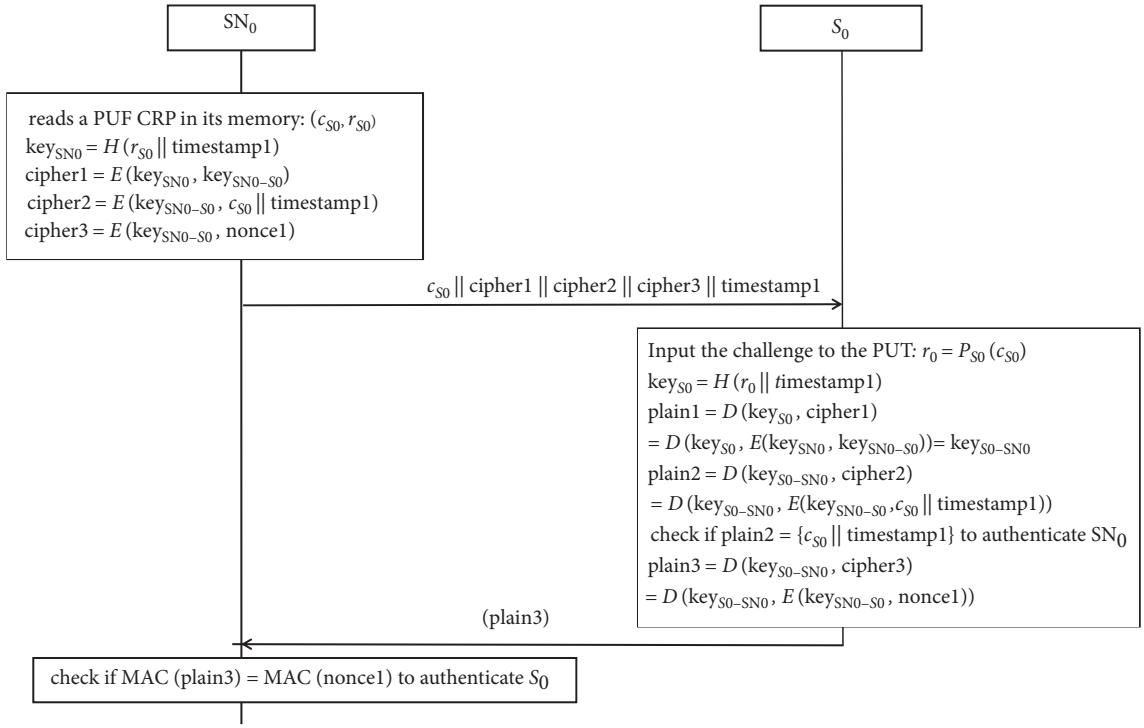

Figure 6: Authentication and key distribution between SN_0 and static sensor S_0.

CRP of SN_0 (c_{S0}, r_{S0}). The SN_0 fails the authentication by S_0 and the scheme quits.

If correct, S_0 infers that $key_{S0-SN0} = key_{SN0-S0;}$ then key_{S0} equals key_{SN0}, and r_0 equals r_{S0}. This means the sink node SN_0 indeed shares a CRP (c_{S0}, r_{S0}) of the PUF P_{S0} and passes the authentication by S_0.

S_0 decrypts the cipher3 by using key_{S0-SN0} and gets plain3:

$$\begin{aligned} \text{plain3} &= D\left(\text{key}_{S0-SN0}, \text{cipher3}\right) \\ &= D\left(\text{key}_{S0-SN0}, E\left(\text{key}_{SN0-S0}, \text{nonce1}\right)\right). \end{aligned} \tag{11}$$

S_0 constructs and sends a message authentication code (MAC) to the SN_0:

$$S_0 \longrightarrow SN_0\text{: MAC (plain3).} \tag{12}$$

(4) SN_0 checks if the equation MAC(plain3) = MAC (nonce1) is correct.

If correct, SN_0 infers that the S_0 carried out a correct nonce1 by computing the correct pairwise key, key_{S0-SN0}, which is derived by the correct response r_{S0} of PUF P_{S0}. Thus, the sensor S_0 passes the authentication by SN_0.

If not, SN_0 deduces that the sensor is not a valid S_0 as it declares, since it cannot output a correct response of r_{S0} so as to compute a correct key_{S0-SN0}. S_0 fails the authentication and quits.

From now on, an intracluster pairwise key $key_{S0-SN0} = (key_{SN0-S0})$ is established and utilized to encrypt the communications between S_0 and SN_0. The mutual authentication is implemented by PUF CRP and the intracluster communication security is assured. Besides, the process is safe from the replay attack because the temporary key is derived involving the timestamps.

3.4. Mobile Sensors Subscheme. The network is dynamic during the working time. As shown in Figure 5, the sensor S_6 moves from the cluster C_2, where it is thrown on, into the cluster region of C_1. Therefore, the Home-SN of S_6 is SN_2 and the Present-SN is SN_1. However, the SN_1 does not share the PUF CRP of S_6, and it should implement the authentication and key distribution via the help of SN_2. The subscheme is described as shown in Figure 7.

(1) The sink node SN_1 broadcasts the id of sensor S_6 to request help. This is a round of intercluster communication.

(2) The sink node SN_2 reads a PUF CRP in its memory: (c_{S6}, r_{S6}) by indexing of id_{S6}.

SN_2 computes a temporary key, key_{SN}:

$$\text{key}_{SN2} = H\left(\left\|r_{S6}\text{timestamp11}\right.\right). \tag{13}$$

SN_2 generates a session key between SN_1 and S_6, key_{SN1-S6}, and encrypts it by key_{SN2}:

$$\text{cipher11} = E\left(\text{key}_{SN2}, \text{key}_{SN1-S6}\right). \tag{14}$$

Then, SN_2 encrypts the c_{S6} by using key_{SN1-S6} to get cipher22:

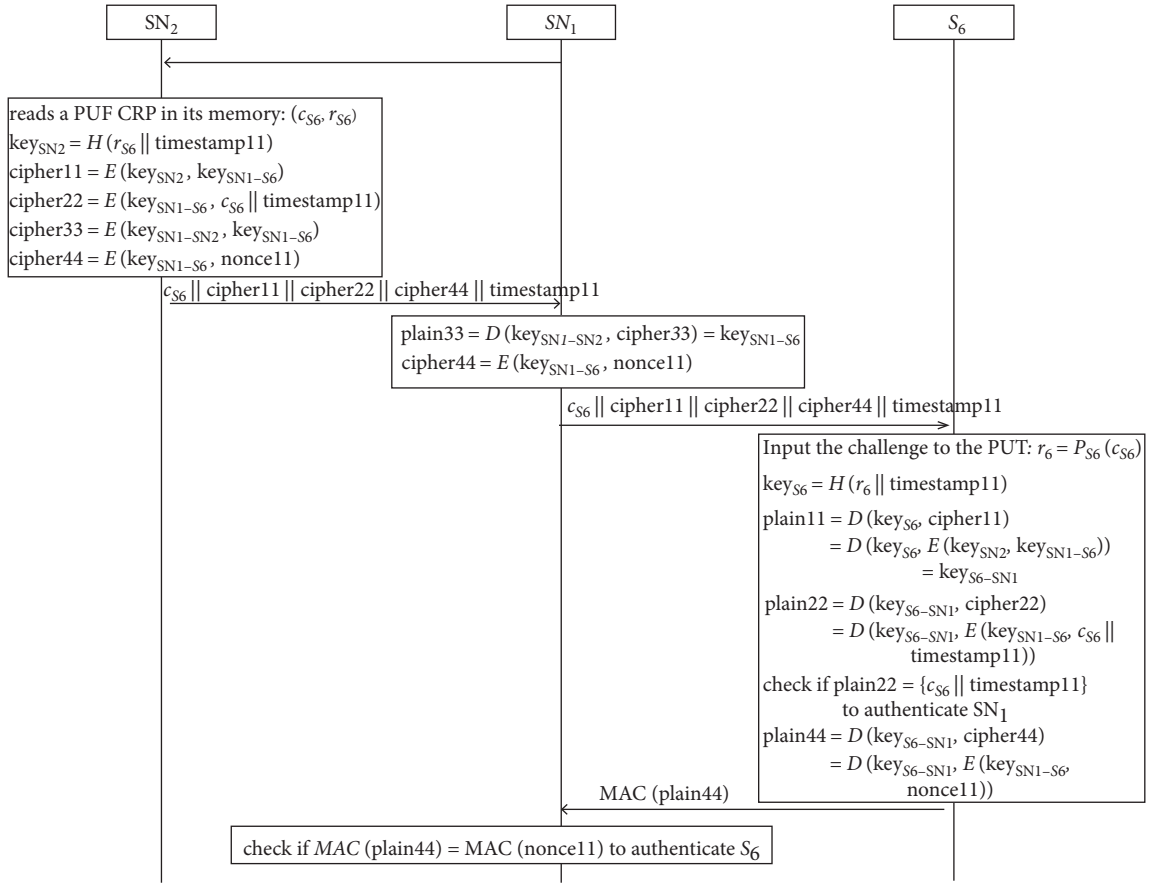

Figure 7: Authentication and key distribution between SN_1 and mobile sensor S_6.

$$\text{cipher22} = E\left(\text{key}_{\text{SN1–S6}}, \|c_{S6}\text{timestamp11}\right). \quad (15)$$

SN_2 encrypts the key $\text{key}_{\text{SN1–S6}}$ by an intercluster key, $\text{key}_{\text{SN1–SN2}}$, shared between SN_1 and SN_2:

$$\text{cipher33} = E\left(\text{key}_{\text{SN1–SN2}}, \text{key}_{\text{SN1–S6}}\right). \quad (16)$$

SN_2 sends the challenge c_{S6}, cipher11, cipher22, and cipher33 to SN_1:

$$\text{SN}_2 \longrightarrow \text{SN}_1\text{: } \|c_{S6}\|\text{cipher11}\|\text{cipher22}\|\text{cipher33timestamp11}. \quad (17)$$

(3) SN_1 decrypts the cipher33 to get the session key, $\text{key}_{\text{SN1–S6}}$:

$$\text{plain33} = D\left(\text{key}_{\text{SN1–SN2}}, \text{cipher33}\right) = \text{key}_{\text{SN1–S6}}. \quad (18)$$

Then, SN_1 generates a secret random number nonce11 and encrypts it by using $\text{key}_{\text{SN1–S6}}$:

$$\text{cipher44} = E\left(\text{key}_{\text{SN1–S6}}, \text{nonce11}\right). \quad (19)$$

SN_1 sends the challenge c_{S6}, cipher11, cipher22, and cihper44 to the sensor S_6:

$$SN_1 \longrightarrow S_6\text{: } \|c_{S6}\|\text{cipher11}\|\text{cipher22}\|\text{cipher44timestamp11}. \quad (20)$$

(4) After receiving the message, the sensor S_6 firstly inputs c_{S0} into the PUF structure P_{S6}, which is embedded during the initialization phase, and gets the output response r_6:

$$r_6 = P_{S6}\left(c_{S6}\right). \quad (21)$$

S_6 computes a temporary key, key_{S6}:

$$\text{key}_{S6} = H\left(\|r_6\text{timestamp11}\right). \quad (22)$$

Then, S_6 decrypts cipher11 to get the pairwise key, $\text{key}_{\text{S6–SN1}}$:

$$\text{plain11} = D\left(\text{key}_{S6}, \text{cipher11}\right) = D\left(\text{key}_{S6}, E\left(\text{key}_{\text{SN2}}, \text{key}_{\text{SN1–S6}}\right)\right) = \text{key}_{\text{S6–SN1}}. \quad (23)$$

S_6 decrypts cipher22 by using $\text{key}_{\text{S6–SN1}}$ and gets plain22:

$$\begin{aligned}\text{plain22} &= D(\text{key}_{S6-SN1}, \text{cipher22}) \\ &= D(\text{key}_{S6-SN1}, E(\text{key}_{SN1-S6}, c_{S6} \| \text{timestamp11})).\end{aligned} \tag{24}$$

The sensor S_6 checks if the equation plain22 = $\{c_{S6} \| \text{timestamp11}\}$ is correct.

If not, S_6 deduces that the cipher11 and cipher22 are not generated from its valid Home-SN or not forwarded from a trusted Present-SN. The SN_1 fails the authentication by S_6 and quits.

If correct, S_6 infers that the Present-SN SN_1 is trusted by SN_2 and passed the authentication.

S_6 decrypts cipher44 by using key_{S6-SN1} and gets plain44:

$$\begin{aligned}\text{plain44} &= D(\text{key}_{S6-SN1}, \text{cipher44}) \\ &= D(\text{key}_{S6-SN1}, E(\text{key}_{SN1-S6}, \text{nonce11})).\end{aligned} \tag{25}$$

S_0 constructs and sends a message authentication code (MAC) to the SN_1:

$$S_6 \longrightarrow \text{SN1: MAC(plain44)}. \tag{26}$$

(5) SN_1 checks if the equation MAC (plain44) = MAC (nonce11) is correct:

If correct, SN_1 infers that the S_6 carried out a correct nonce11 by computing the correct pairwise key, key_{S6-SN1}, which is derived by the correct response r_{S6} of PUF P_{S6}. Thus, the sensor S_6 passes the authentication by SN_1.

If not, SN_1 deduces that the sensor is not a valid S_6 as it declares, since it cannot output a correct response of r_{S6} to compute a correct key_{S6-SN1}. S_6 fails the authentication and quits.

4. Simulation, Analysis and Comparisons

We present the security and performance evaluation of the proposed scheme through simulation experiments and analysis. We provide extensive simulations to verify the performance metrics such as secure connectivity, resilience against node capture, memory consumption, and communication overhead. We compare the proposed approach with other key management schemes. In the simulation, we assume 10000 sensor nodes, and 100 sink nodes are randomly distributed in a 1000×1000 m field. Each sensor node has a fixed speed ranging from 1 to 10 m/s. The radio range of each sensor node is considered as 50 m.

4.1. Mutual Authentication. The basic idea of the authentication of our approach is the challenge-response mechanism based on the PUF CRP. In both subschemes, mutual authentication between the sink node and the (static or mobile) sensor is assured. Furthermore, the scheme quits before key distribution process if the authentication failed, that is, an unauthenticated sensor cannot participate the whole communication network. Compared with the PKI method, the PUF-based authentication speeds up and reduces the storage requirement.

In some proposed PUF authentication schemes [21, 22], the challenge and response are always sent in plaintext. If attackers catch an entire PUF CRP, they are able to launch the replay attack and man-in-the-middle attack. In order to resist the replay attack, a strong PUF is usually employed to provide a plenty of CRPs and each of them is only used once. Then, different CRPs of a PUF are openly exposed in a dynamic network where a mobile node needs frequent authentication with new neighbors. This PUF structure is vulnerable to the modelling attack that tries to guess and predict the response value related to a certain challenge.

In our scheme, the PUF response is not transmitted in plain but converted into an encryption key by hashing with a timestamp. A node succeeds the authentication if it decrypts and carries out a correct plaintext. This is a kind of symmetric authentication [34] combined with the PUF challenge-response mechanism. In order to prevent the replay attack, a timestamp has been used. The fact that the PUF response is not transmitted in plain effectively resists the modelling attack on PUF.

4.2. Overheads. We mainly consider the energy consumption in terms of storage, communication, and computation overheads. We mainly consider the following assumptions: MAC size is considered as 4 bytes, 4 bytes for time stamp, random nonce as 16 bytes, 32 bytes for key size, and 32 bytes for challenge/response of a PUF. We also consider 2 bytes for the node ids. The ciphertext has the same length with the key.

4.2.1. Key Storage. In our approach, during the initialization phase, each sensor is not predistributed with any key in its memory, while each sink node is predistributed with n/m PUF CRPs. A PUF structure is embedded in a sensor (as a hardware) during the initialization phase (therefore, the storage overhead is not discussed in this paper). After the key distribution, the sensor stores 1 intracluster session key established with the sink node, while the sink node stores one intracluster session key for each cluster-member sensor. All the intermediate data generated in the key distribution process is deleted to release the storage space. Therefore, the storage overhead of a sensor is 32 bytes and that of a sink node is $(32 + 32 \times 2 + 2)n/m = 98n/m$ bytes.

Du et al. proposed an AP scheme [10], which is a pure random key predistribution scheme. The main idea is to preload only a small number of keys (denoted as l) in low-ended sensors, while preloading a relatively large number of keys (denoted as $M \gg l$) in each high-ended sink nodes. Any two nodes cannot establish a secure link if they do not share a common pairwise keys. Therefore, nodes need to store more keys to increase the probability of sharing common keys, which is defined as the secure connectivity. As analyzed in Erfani's scheme [16], the sensor memory is partitioned into two parts: store α predistributed keys in the first part and β postdeployment keys in the second part. Each pair of

neighboring nodes establish a common predistributed or postdeployment key to secure the communication. Erfani's scheme claimed that each sink node stores only 1 key; BS stores a key table, which contained some information about sensor nodes' keys. In addition, BS is aware of sink nodes' keys.

Table 1 compares the amount of memory required for storing keys in the proposed scheme and other two solutions. The key storage in sink node of our scheme is higher than Erfani's scheme, but the storage of sensor is much lower than both Erfani's and AP schemes. Therefore, our scheme is efficient for resource limited sensor nodes, and this performance also brings an advantage of better resilience against node capture attack.

4.2.2. Communication Overhead. In this paper, the communication overhead is measured by the message size and transmission rounds but does not consider the message overhead consisting of a protocol ID, a message ID, a checksum, and the headers and footers of the low-level network layers.

We analyze the communication overhead for static and mobile subschemes, respectively.

In the static subscheme, to establish an intracluster pairwise key, the sensor sends only 1 MAC packet with 4 bytes, while the sink node sends 1 packet with 132 bytes.

In the mobile subscheme, to establish an intracluster pairwise key, the sensor sends only 1 MAC packet with 4 bytes, while the Home-SN sends 1 packet with 132 bytes and the Present-SN sends 2 packets with 2 bytes and 132 bytes.

Compared with the random key predistribution schemes like the AP, nodes do not need key construction or authentication but try to find a common key by sending the key indexes or encrypted challenges. The transmitted message size is linearly related to the size of the keyring. However, if two neighboring nodes do not share a common key, they must send further messages to ≥2 hops intermediate nodes.

4.2.3. Computation Overhead. The most computation overhead is related to cryptography and authentication operations, and the PUF computation especially for sensors. As shown in Table 2, to establish an intracluster pairwise key, the number of encryption or decryption operations in each sensor is 3 and 3 or 5 in a sink node. All these schemes use light weight cryptography methods. The computation overhead is higher than the random key predistribution scheme AP but still acceptable for both sensors and sink nodes.

4.3. Secure Connectivity. The security connectivity of a network is defined as the probability that two entities can establish a session key to secure the communications. Since this paper mainly proposes an approach for intracluster authentication and key distribution, we define the conception of "intracluster secure connectivity" as the probability that a sink node can establish a pairwise key with a cluster-member (static or mobile) sensor.

Table 1: Comparison of storage overhead in different schemes (bytes).

	Our scheme	Erfani's	AP
Sensor	32	$32(\alpha+\beta)$	$32l$
Sink	$98n/m$	32	$32M$
BS	—	$32[n(\alpha+\beta)+m]$	—

Table 2: Comparison of computation overhead.

	Our scheme	AP
Cryptography in sensor	3	NA
Cryptography in sink node	Static: 3 Mobile: 5	NA
PUF	1	NA

This scheme is a kind of deterministic key distribution model, in which any sensor node can successfully establish a session key with no matter the Home-SN or the Present-SN. Therefore, the intracluster security connectivity is 100% in this scheme, which is a remarkable improvement compared with the probabilistic key distribution schemes [9, 10, 12].

The random schemes, like AP scheme, must increase the amount of key storage to achieve high security connectivity. Figure 8 shows the secure connectivity versus the key pool size P in the AP. There are four solid curves in Figure 8, from bottom to top, corresponding parameters $[l, M]$ of [5, 125], [10, 250], [15, 375], and [20, 500], respectively. It is observed that the probability of sharing key increases when the number of preloaded keys increases. For the same parameters $[l, M]$, the probability of sharing key decreases as the key pool size becomes large. In Figure 9, we also plot the secure connectivity for different numbers of preloaded keys in the AP and our scheme. As analyzed in the above section, the storage overhead of the sink node in our scheme is $98n/m \approx 10000$ bytes, almost 300 32bytes-keys. It is worth emphasizing that the key storage of sensor nodes in our proposal is 0, which is significantly lower than that of AP scheme, but the connectivity is significantly higher than that of AP scheme. The Erfani's scheme is also claimed of providing full secure connectivity in [16], however there is a tradeoff between α and β in balancing the storage, connectivity, and resilience.

4.4. Resilience Against Node Capture. Sensor networks are usually deployed in an unattended environment, and attackers illegally obtain the secret information of nodes by capturing nodes and other physical attacks. Resilience against node capture is defined as the probability $F(x)$ that the attacker can obtain the key in the uncaptured node directly or indirectly according to a certain number of captured nodes x:

$$F(x) = \frac{\text{number of compromised links between uncaptured nodes}}{\text{number of uncompromised links}}. \quad (27)$$

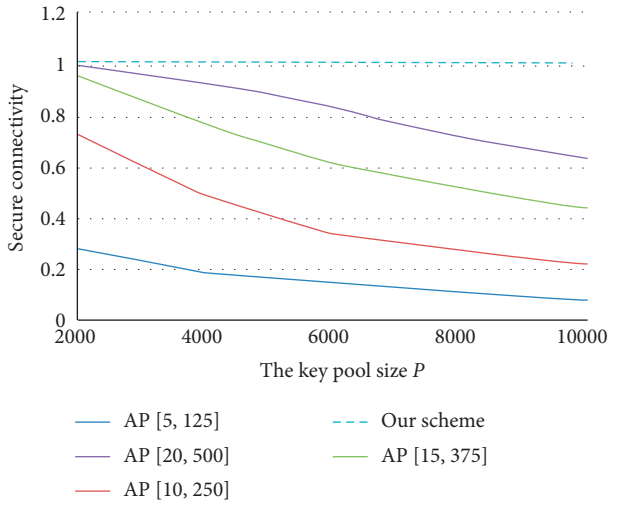

Figure 8: Secure connectivity versus the key pool size P.

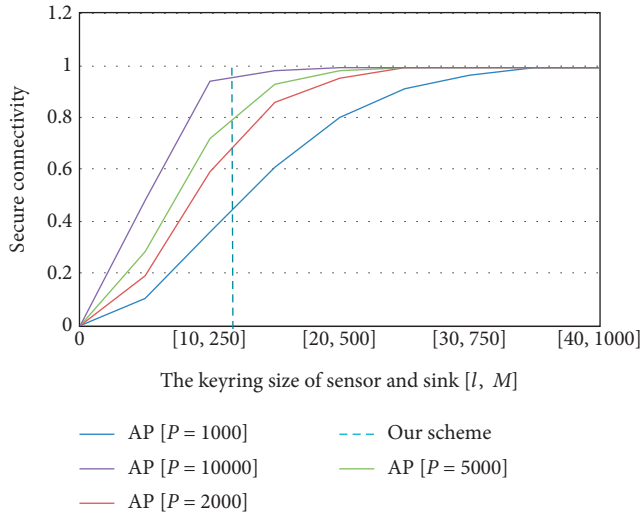

Figure 9: Secure connectivity versus keyring size.

4.4.1. Resilience against the Sensor Capture. Different from the traditional random key predistribution schemes [4, 5, 7], in this scheme, the sensor node does not prestore any keys or other key materials, which not only reduces the storage cost of the sensor but also improves the resilience against the sensor capture because the attacker cannot obtain any key that belong to a safe node despite capturing a sensor physically. Therefore, our proposal has perfect resilience against the sensor capture; that is,

$$F(xS) = 0, \tag{28}$$

where x_S represents the number of captured sensor nodes.

4.4.2. Resilience against the Sink Node Capture. The sink node acts as the cluster head, which maintains the intracluster secure communication with the cluster members and also the intercluster secure communication with other cluster heads externally. Each sink node is prestored with a number of CRPs in the initialization phase and uses the CRPs to authenticate and distribute pairwise keys with its cluster-member sensors.

The physical capturing of a sink node breaks up both the internal and external cluster communication of it. The dismissed cluster members (sensors) become isolated nodes and may join other clusters. By repeating the authentication and key distribution process, the dismissed sensor obtains a new session key with its new cluster head. There is not any key that belong to a safe node that will be exposed by a physical captured sink node. Therefore, our proposal has perfect resilience against the sink node capture; that is,

$$F(x_{\mathrm{SN}}) = 0, \tag{29}$$

where x_{SN} represents the number of captured sink nodes.

4.4.3. Resilience against Selective Node Capture. Huang et al. [35] pointed out that, in many key management schemes, the selective node capture causes more damage to the network. In the selective node capture attacks, attackers attempt to capture nodes that may reveal more valid and fresh information about uncaptured nodes. In our proposed scheme, an adversary cannot figure out which sink node owns the CRP of a certain sensor, because all CRPs are randomly and safely selected from the CRP pool. Therefore, unless the adversary compromises all the sink nodes, it cannot choose a certain sink node to capture to maximize the uncompromised keys.

4.4.4. Simulation Results. The AP scheme [10] proposed by Du et al. is a pure random key predistribution scheme in cluster sensor networks, with the advantage in saving nodes' communication and computation overheads. But it is hard to balance the tradeoff between the security connectivity and security. Boujelben et al. [12] improved the AP by combining the Blom matrix in terms of the resilience against node capture but require quantity of storage overhead for matrix parameters. Erfani's scheme [16] is a combination of the key pre-distribution and post-deployment key management scheme. When a sensor is captured, all predistributed and postdeployment keys of the node are compromised. But since the postdeployment key is not selected form the key pool, the compromise of such key does not affect the security of other communications, whereas compromising the predistributed keys of a sensor node will make other communication links insecure, because such keys are selected from the key pool and might be common with some sensors. Erfani's scheme provides better resilience against node capture attack than the AP, and the resilience of sensor network depends on the number of predistributed keys α and key pool size P.

We will compare our scheme with these schemes by simulation experiments. The size of key pool in AP, Boujelben's, and Erfani's schemes is $P = 10000$. Similar to the experiments environment in [16], the keyring size is 100 in Erfani's scheme.

As shown in Figures 10 and 11, the experimental results prove that, in the random key predistribution schemes, the resilience against node capture gets worse and worse with the number of captured nodes increasing, because the nodes store a large number of keys. In Boujelben et al. scheme, the nodes store matrixes instead of keys, so the resilience against node capture is better than that in the AP scheme, but the storage cost is λ times that of AP (λ is the matrix parameter).

In our scheme, the sensor node does not store any key, and the sink node stores the CRPs rather than the key as well, so perfect resilience against node capture is provided.

4.5. PUF Security. In this paper, PUF is the core of the authentication and key distribution. The security of the PUF is crucially important. The main threats to some PUF-based schemes [36] include man-in-the-middle attack, replay attack, and the modelling attack to the PUF, because the PUF CRPs are transmitted in plain form. A PUF is considered failed when the adversaries can guess more than 75% bits of the response to a challenge after obtaining enough amount of CRPs of a given PUF. In our proposal, the response, generated by a PUF on a sensor on-the-fly, is not sent to the sink node directly but is utilized as an encryption key to encrypt the challenge. Such design can successfully protect the PUF from cloning attack, modelling attack, and side channel attacks, including electromagnetic analysis attack and differential fault attack. The eavesdropping is invalid, since all the transmitted messages are encrypted with symmetric algorithm (e.g., AES), the attackers cannot get any plain information about responses or keys. The scheme can withstand the man-in-the-middle attack and tamper attack, since the encrypted response protects its integrity in the wireless communications.

In the replay attack, an attacker resends an old message, which has been sent for key generation request. In the proposed approach, timestamp has been used in generating the temporary key to prevent the replay attack. Besides, the session key is randomly generated between the sink node and sensor and will not be the same as the a priori key. An attacker can continuously resend an old message to consume the energy of sensor nodes; however, these messages will be discarded.

Table 3 shows the comprehensive comparison results among different authentication and key distribution schemes for sensor networks proposed in recent years. Unlike the key predistribution schemes, for example, AP [10], our scheme is perfectly resilient against node capture attacks, because a sensor does not prestore any keys that might secure other sensors' communications. PUF CRPs provide a type of authentication by a challenge-response mechanism, but Chatterjee's scheme [23] does not guarantee mutual authentication between two parties. In addition, PUFs provide another type of security guarantee implied by their unclonability and tamper evidence. Such property is only available to PUF-based solutions. However, PUF CRPs are sent as plaintext in [23, 25], which make them vulnerable to impersonation attack, but we avoid this in our scheme by encrypting the response of the CRPs. Also, in [14, 23, 25, 27], they used public key algorithm that consumed more computation overhead than the AP [10] and our proposal.

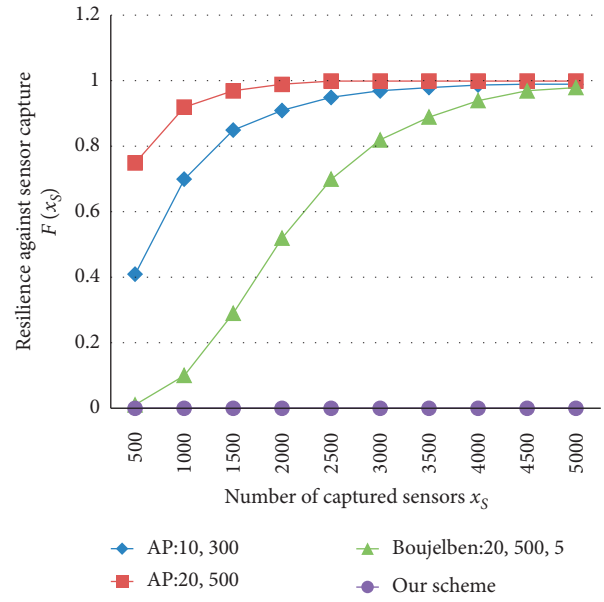

Figure 10: Resilience against the sensor capture.

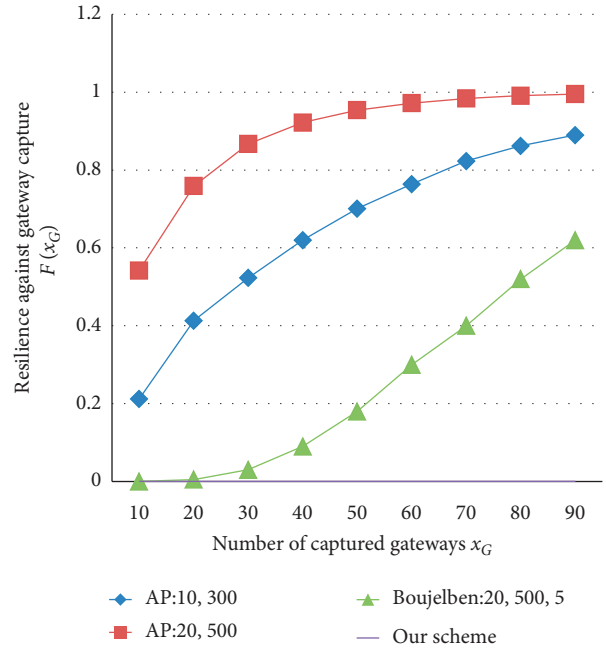

Figure 11: Resilience against the sink node capture.

Table 3: Comparisons of different key distribution schemes.

Property	Ours	AP [10]	Lee and Kim [14]	Erfani et al. [16]	Chatterjee et al. [25]	Li et al. [27]
Public key encryption	No	No	Yes	No	Yes	Yes
Key predistribution	No	Yes	No	Yes	No	No
Perfect resilience against node capture	Yes	No	—	No	—	—
PUF-based	Yes	No	No	No	Yes	Yes
Mutual authentication	Yes	No	Yes	Yes	No	Yes
Resistant to modelling attacks	Yes	—	—	—	No	No
Resistant to eavesdropping attacks	Yes	—	—	—	Yes	Yes
Resistant to collusion attacks	Yes	—	—	Yes	—	—

—: not applicable.

5. Conclusions

In a dynamic sensor network, how to ensure two communicating (static or mobile) nodes authenticate and share a pairwise key is difficult because the sensors' mobility pattern or track is unknown. In this paper, we propose a mutual-authenticated key distribution scheme for the intracluster communication. In order to reduce the storage overhead and the key exposure risk of low-end sensors, we employ a CRO Physical Unclonable Function (PUF) in the mutual-authentication process, which has the lightweight, unclonability, and unpredictability advantages. Compared with the classical PUF challenge-response authentication mechanism in some literatures, the PUF response is not transmitted in plain forms so as to resist the modelling attacks on PUFs. We also demonstrate that the proposed scheme improves the secure connectivity and other performances by analysis and experiments.

Data Availability

The data are available at https://www.zhangqiaokeyan.com/patent-detail/06120103885959.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (under Grant 61902163), the Research Startup Foundation of Jinling Institute of Technology (under Grant JIT-B-201639), and the Key Program of National Key Research and Development Project "Cybersecurity" (under Grant 2017YFB0802800).

References

[1] D. Carman, P. Kruus, and B. Matt, *Constraints and approaches for distributed sensor network security (final)*, pp. 1–139, NAI Labs Technical Report, NAI Labs, MD, USA, 2000.

[2] Y. Ren, Y. Leng, J. Qi et al., "Multiple cloud storage mechanism based on blockchain in smart homes," *Future Generation Computer Systems*, vol. 115, pp. 304–313, 2021.

[3] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An AI-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of IoT," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2021.

[4] G. Liu, Q. Yang, and H. Wang, "Trust assessment in online social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 2, pp. 994–1007, 2018.

[5] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia, and L. Fang, "Revocable attribute-based encryption with data integrity in clouds," *IEEE Transactions on Dependable and Secure Computing*, vol. 21, no. 3, p. 1, 2021.

[6] C. Ge, W. Susilo, Z. Liu, J. Xia, P. Szalachowski, and F. Liming, "Secure keyword search and data sharing mechanism for cloud computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 4, p. 1, 2020.

[7] C. Y. Chen and H. C. Chao, "A survey of key distribution in wireless sensor networks," *Security and Communication Networks*, vol. 7, 2014.

[8] D. Farooq and M. Gull, "A survey about applications, issues and challenges of sensor network," *International Journal of Computer Applications*, vol. 180, no. 19, pp. 41–46, 2018.

[9] L. Eschenauer and V. D. Gligor, "A key management scheme for distributed sensor networks," in *Proceedings of the ACM Conference on Computer and Communication Security*, pp. 41–47, Washington, DC, USA, November 2002.

[10] X. Du, Y. Xiao, M. Guizani, and H.-H. Chen, "An effective key management scheme for heterogeneous sensor networks," *Ad Hoc Networks*, vol. 5, no. 1, pp. 24–34, 2007.

[11] R. Blom, "An optimal class of symmetric key generation systems," *Advances in Cryptology-Eurocrypt*, vol. 84, pp. 335–338, 1984.

[12] M. Boujelben, O. Cheikhrouhou, M. Abid, and H. Youssef, "Establishing pairwise keys in heterogeneous two-tiered wireless sensor networks," in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications Athens*, pp. 18–23, Athens, Greece, 2009.

[13] K. Benamar, F. Mohammed, and M. Abdellah, "Architecture aware key management scheme for wireless sensor networks," *International Journal of Information Technology & Computer Science*, vol. 4, no. 12, pp. 50–59, 2012.

[14] S. Lee and K. Kim, "Key renewal scheme with sensor authentication under clustered wireless sensor networks," *Electronics Letters*, vol. 51, no. 4, pp. 368-369, 2015.

[15] K. Han, K. Kim, and T. Shon, "Untraceable mobile node authentication in WSN," *Sensors*, vol. 10, no. 5, pp. 4410–4429, 2010.

[16] S. H. Erfani, H. H. S. Javadi, and A. M. Rahmani, "A dynamic key management scheme for dynamic wireless sensor networks," *Security and Communication Networks*, vol. 8, no. 6, pp. 1040–1049, 2015.

[17] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in

DWSNs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.

[18] J. Delvaux, R. Peeters, D. Gu, and I. Verbauwhede, "A survey on lightweight entity authentication with strong PUFs," *ACM Computing Surveys*, vol. 48, no. 2, pp. 1–42, 2015.

[19] A. R. Sadeghi and D. Naccache, "Towards hardware-intrinsic security," *Information Security & Cryptography*, vol. 364, no. 1849, pp. 3215–3230, 2010.

[20] B. Gassend, D. E. Clarke, and M. V. Dijk, "Silicon physical random Functions," in *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 18–22, Washington, DC, USA, November 2002.

[21] P. Tuyls, G. J. Schrijen, and B. Skoric, "Read-proof hardware from protective coatings," in *Proceedings of the 8th International Workshop of Cryptographic Hardware and Embedded Systems—CHES 2006*, Yokohama, Japan, October 2006.

[22] C. Brzuska, M. Fischlin, H. Schröder, and S. Katzenbeisser, "Physically uncloneable functions in the universal composition framework," *Advances in Cryptology—CRYPTO 2011*, vol. 6841, pp. 51–70, 2011.

[23] A. Allam, "FPGA-based authenticated key exchange scheme utilizing PUF and CSI for wireless networks," in *Proceedings of the IEEE International Conference on System of Systems Engineering (SoSE 2015)*, pp. 170–175, Monterey, CA, USA, 2015.

[24] R. Bahrampour and R. E. Atani, "A novel key management protocol for wireless sensor networks based on PUFs," *International Journal of Future Generation Communication & Networking*, vol. 6, no. 2, pp. 93–106, 2013.

[25] U. Chatterjee, R. S. Chakraborty, and D. Mukhopadhyay, "A PUF-based secure communication protocol for IoT," *ACM Transactions on Embedded Computing Systems*, vol. 16, no. 3, pp. 1–25, 2017.

[26] A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, no. 8, pp. 1–15, 2018.

[27] S. S. Li, Y. C. Huang, and B. Yu, "A PUF-based low cost secure communication scheme for IoT," *Acta Electronica Sinica*, vol. 47, no. 04, pp. 46–51, 2019.

[28] Z. Zhang, Y. Liu, Q. Zuo, L. Harn, S. Qiu, and Y. Cheng, "PUF-based key distribution in wireless sensor networks," *Computers, Materials & Continua*, vol. 64, no. 2, pp. 1261–1280, 2020.

[29] Y. Cui, C. Wang, and W. Liu, "Low-cost configurable ring oscillator PUF with improved uniqueness," in *Proceedings of the IEEE International Symposium on Circuits & Systems*, pp. 558–561, IEEE, Baltimore, MD, USA, 2016.

[30] L. Zhang, C. H. Wang, W. Q. Liu, M. O'Neill, and F. Lombardi, "XOR Gate Based Low-Cost Configurable RO PUF," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4, Baltimore, MD, USA, 2017.

[31] Y. Cui, C. Gu, C. Wang, M. O'Neill, and W. Liu, "Ultra-lightweight and reconfigurable tristate inverter based physical unclonable function design," *IEEE Access*, vol. 6, pp. 28478–28487, 2018.

[32] A. Maiti and P. Schaumont, "Improved ring oscillator PUF: an FPGA-friendly secure primitive," *Journal of Cryptology*, vol. 24, no. 2, pp. 375–397, 2011.

[33] Y. Mohamed, Y. Moustafa, and A. Khaled, "Energy-aware management for cluster-based sensor networks," *Computer Networks*, vol. 43, no. 5, pp. 649–668, 2003.

[34] S. Malhotra and M. C. Trivedi, "Symmetric key based authentication mechanism for secure communication in MANETs," in *Intelligent Communication and Computational Technologies*Springer, Singapore, 2018.

[35] D. Huang, M. Mehta, D. Medhi, and L. Harn, "Location-aware key management scheme for wireless sensor networks categories and subject descriptors," in *Proceedings of the Second ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 29–42, Washington, DC, USA, October 2004.

[36] M. N. Aman, K. C. Chua, and B. Sikdar, "Position paper: physical unclonable functions for IoT security," in *Proceedings of the ACM international workshop*, pp. 10–13, ACM, Seoul, Republic of Korea, 2016.