Hindawi Publishing Corporation
Journal of Sensors
Volume 2016, Article ID 1321079, 11 pages
http://dx.doi.org/10.1155/2016/1321079

Hindawi

Research Article

An Energy-Efficient Secure Scheme in Wireless Sensor Networks

Kyoungsoo Bok,[1] Yunjeong Lee,[2] Junho Park,[3] and Jaesoo Yoo[1]

[1]*School of Information and Communication Engineering, Chungbuk National University, 1 Chungdaero, Seowon-gu, Cheongju 362-763, Republic of Korea*
[2]*LOTTE Engineering & Construction IS Team, LOTTE Data Communication Co., Saerom B/D, No. 18-1, Jamwon-dong, Seocho-gu, Seoul 137-903, Republic of Korea*
[3]*Agency for Defense Development, Bugyuseong-daero 488, Yuseong, Daejeon 305-600, Republic of Korea*

Correspondence should be addressed to Jaesoo Yoo; yjs@chungbuk.ac.kr

Received 18 November 2015; Revised 28 April 2016; Accepted 10 May 2016

Academic Editor: Hana Vaisocherova

We propose an energy-efficient security scheme in wireless sensor networks. The proposed scheme converts sensing data using TinyMD5, which is a variation of MD5, a one-way hash function, and can solve the collision problem of hash value that occurs when MD5 is modified. In addition, it strengthens security capabilities by transmitting data through multiple paths after conversion with TinyMD5 and divides the data to make decryption of the original data difficult. To show the superiority of the proposed algorithm, we compare it with the existing schemes through simulations. The performance evaluation results show that the proposed scheme maintains security better than the existing scheme, improving the communication cost and the network lifetime.

1. Introduction

A wireless sensor network is one of the core next-generation application fields. It has been applied in a variety of fields, such as ecosystem monitoring, military zone surveillance, and U-City applications. A wireless sensor network consists of multiple sensor nodes that can perform environment information collection, computation, and wireless communication. A wireless sensor network collects environment information by utilizing sensor modules and analyzes collection information with various applications [1–5]. However, the performance of sensor nodes in sensor networks is limited in terms of energy, computational capability, communication radius, and storage memory. Therefore, the algorithms of all wireless sensor network schemes should be studied to minimize energy consumption by utilizing energy efficiently and considering the limited performance of sensor nodes [6].

Wireless sensor networks are vulnerable to external attacks, as they are constructed mostly in unmanned environments to monitor environmental information or military zones, and sensor nodes use wireless communication to transfer collected data to base stations [7–9]. Due to such characteristics, data can be easily exposed during data transmission. This can be a serious problem in applications that deal with military information or individual privacy. To solve these problems, studies on security schemes have been performed.

To configure a safe sensor network, it is possible to combine a routing protocol, which is a basic element for information transmission, with an encryption protocol, such as key exchange and authentication [10–12]. Recently, the various data transmission schemes were proposed such as a transmission scheme that periodically collects real and virtual data at the same time based on encryption algorithms and a data transmission scheme that uses reliability levels [13, 14]. However, such schemes are not suitable for wireless sensor nodes with performance limitations, such as energy consumption due to additional data transmission or increased communication between nodes for increasing security. Therefore, it is necessary to study energy-efficient security schemes while considering the characteristics of wireless sensor networks. Therefore, this paper aims to provide an algorithm to restrict data analysis of transferred data over a wireless sensor network despite transferred data exposure as well as minimize energy consumption for communication and improve security and energy efficiency.

In this paper, we propose an energy-efficient security scheme, called TinyMD5, which modifies MD5 [15], for wireless sensor networks. TinyMD5 is a one-way hash function for restricting data analysis despite data exposure attempts from external attacks during the data transmission process over a wireless sensor network. In addition, we propose a data transmission scheme in which transformed data are divided and transmitted to avoid full data exposure, which enables data analysis. The basic concept of the proposed scheme was developed based on the fact that it is impossible to identify full data information using only partial data, and there is no decryption method for MD5, a one-way hash function. Accordingly, it converts data using TinyMD5, which is a variation of the MD5 algorithm, and transmits data into multiple paths by dividing converted data. The proposed scheme transfers data via GPSR [16], which can increase security and minimize the communication between sensors.

The rest of this paper is organized as follows. In Section 2, relevant foundational knowledge and the characteristics of existing security schemes are analyzed to explain past problems. In Section 3, the proposed energy-efficient secure transmission scheme for wireless sensor networks is described. In Section 4, the effectiveness of the proposed scheme is assessed through a performance evaluation. In Section 5, the conclusion of this study and suggestions for future research are presented.

2. Related Work

DES [17] and AES [18] are used for data encryption in sensor networks. Recently, data encryption using one-way hash function such as MD5 [15] is used. DES is a symmetric-key algorithm for data encryption. DES key consists of 64 binary digits, where 56 bits is randomly generated and the remaining 8 bits may be used for error detection [19]. DES has been superseded by AES. The AES is a specification for the encryption of electronic data established by the US National Institute of Standards and Technology. AES is a variant of Rijndael which has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits [20]. The MD5 algorithm is an extension of the MD4 message-digest algorithm. The MD5 is a widely used cryptographic hash function producing a 128-bit hash value. MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks and the message is padded so that its length is divisible by 512 [21].

Karlof et al. introduced a link layer security architecture called TinySec in wireless sensor networks [22]. TinySec supports two different security options: authenticated encryption (TinySec-AE) and authentication only (TinySec-Auth). With authenticated encryption, TinySec encrypts the data payload and authenticates the packet with MAC. The MAC is computed over the encrypted data and the packet header. In authentication-only mode, TinySec authenticates the entire packet with MAC, but the data payload is not encrypted. TinySec uses a specially formatted 8-byte IV and cipher block chaining to provide secure encryption.

Lu et al. proposed TESP^2 which consists of three phases such as system initialization phase, sensor nodes deployment phase, and privacy-preservation sensor data report phase for preserving source privacy in wireless sensor networks [13]. Each sensor node broadcasts timed data collection request to its upstream nodes, and then every upstream node will return the real data's ciphertext if it has sensed something or a dummy data's ciphertext if it has not. After receiving all messages from its upstream nodes, the cluster head checks the validity of each message's MAC and then further filters the dummy messages. The cluster head aggregates a group of reencrypted ciphertexts. To preserve the source privacy, cluster head does not immediately send them to its downstream node until the downstream node broadcasts a data collection request.

Samundiswary et al. proposed S-GPSR that incorporates a trust based mechanism in the existing greedy perimeter stateless routing (GPSR) protocol in mobile sensor networks [14]. S-GPSR used a trust mechanism and mobility model for nodes in GPSR to protect nodes from sinkhole attacks. GPSR scans its neighborhood table and chooses an adjacent neighbor that has the least distance to a particular destination to select the next hop for routing [16]. S-GPSR contains the trust level in the neighborhood table to generate the most trusted distance route rather than the default minimal distance. To compute direct trust in a node, an effort-return based trust model is used. The Trust Update Interval (TUI) of forwarded packet is buffered in the node. It determines the time a node should wait before assigning a trust or distrust level to a node.

Li and Ren proposed a routing through a random intermediate node (RRIN) providing source location privacy through a single randomly selected intermediate node away from the source node [23]. RRIN transfers the message to the randomly selected intermediate node(s) before it is transmitted to the sink node. The intermediate node is expected to be away from the source node for a minimum distance and then RRIN provide great local source location privacy. However, it may not be able to provide adequate global source location privacy. RRIN proposed the routing scheme routing through multiple randomly selected intermediate nodes based on angle and quadrant to improve the global source location privacy.

Duan et al. analyzed features of common attacks on trust-aware routing schemes and proposed a trust-aware secure routing framework (TSRF) to satisfy the security requirements of routing protocols in wireless sensor networks [11]. TSRF used a lightweight computation method to evaluate the trust value of sensor nodes and designed the trust computation of paths to calculate the trust value of the route when packets are transmitted to a destination node via multihop route. The optimal routing path is selected by the trust metric and QoS requirements.

3. The Proposed Secure Transmission Scheme

3.1. Characteristics. In this paper, we propose an energy-efficient secure transmission scheme for wireless sensor

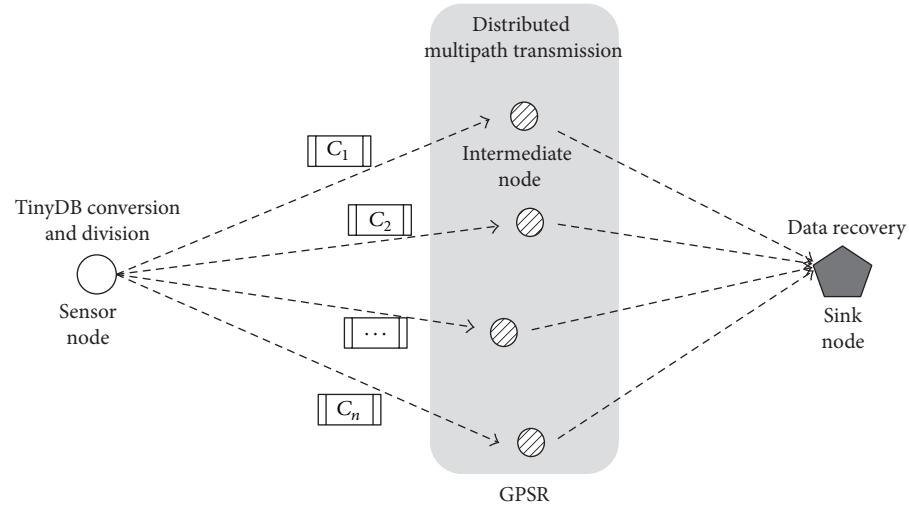

Figure 1: Process of the proposed scheme.

networks. The proposed scheme is characterized by the simplification of the hash function output into 32 bits so that the communication between the sensor nodes increases as the size of the data increases and the amount of numeric data generated in a general wireless sensor network is not too large. In addition, recovery computation is minimized by adding range information for attribute values. Furthermore, since original data can be inferred from full hash values, data is divided and transmitted into different paths to prevent full data loss and solve the problem of original data inference.

The proposed scheme consists of three steps as shown in Figure 1. The first step is to process data in a sensor node, in which data are converted using TinyMD5, a variation of MD5, and conduct the preprocessing step required for distributed data transmission. The second step is to transmit data to the base station by a sensor node, in which the divided data are transmitted with GPSR via an intermediate node. The third step is a data processing step in the base station, in which the original data are recovered from the collected divided data.

3.2. TinyMD5 Conversion and Division. In this paper, the data transmission security has been strengthened through data conversion using a one-way hash function, which has no known decryption algorithm, over a wireless sensor network. Hash functions produce code value outputs that have no regular patterns, which is why it is impossible to identify original data values concretely. In addition, because data sizes (except for multimedia data) are relatively small over a wireless sensor network, this paper proposes an energy-efficient hash algorithm, called TinyMD5, for data security over a wireless sensor network, which was developed by modifying and reducing MD5 in accordance with the characteristics of wireless sensor networks. The proposed TinyMD5 minimizes data transmission costs from encrypted data transmission by scaling down the lengths of hash values by a quarter while maintaining the advantages of original data information protection and the nonlinear pattern creation of existing MD5 hash values. The proposed algorithm inputs basic arbitrary lengths of data and processes it into a 128-bit block, thereby outputting a 32-bit hash value. As shown in Figure 2, the TinyMD5 algorithm consists of three steps. The first step is data initialization, which is conducted to process the data used in the next step. The second step is a block data processing, which processes 128-bit block data according to the predetermined operation. The third step is a range information addition, in which the range information for original data recovery is added to the converted data.

The data initialization step is a preparation process to process the data as a 128-bit block, in which the data is converted into a multiple of 128 bits. Figure 3 shows an example of the data initialization step. The characters in the original data are converted into binary values based on the ASCII code. To convert data into a multiple of 128 bits, once the length of the data converted into a binary value is calculated, the original data as well as padding bit is distinguished by adding "1" to the end of the original data, followed by filling the remaining bits with "0" to pad them out to 112 mod 128. Note that an original data length is inserted into the last 16 bits.

Once padding bit additions and data length insertions are finished, MD buffer values are initialized for the next step of the operation. The MD buffer is used to store the intermediate and final results of the block data that is computed in the block data processing step, which is represented as four 8-bit registers (A, B, C, and D). Initialized data are processed computed in the block data processing step to produce a 32-bit output value. The previously padded data are input and computed as a 128-bit block unit, along with an 8-bit MD buffer value, as well as 16 elements consisting of integer parts of $0.703125(i) * 81349 \bmod 256$. Figure 4 shows the operation structure of the data processing step. A, B, C, and D have 8-bit MD buffer values of 0x8a, 0xdb, 0x75, and 0x24, respectively. $X[k]$ $(k = 0, 1, \ldots, n)$ is the kth block data obtained by dividing data block that is processed in the data initialization to the 32-bit data. The input order of $X[k]$ can be different in every round, and ordering is

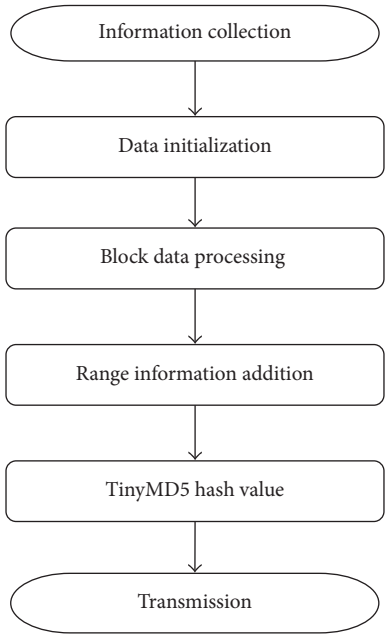

Figure 2: TinyMD5 algorithm.

Table 1: TinyMD5 irreducible function.

Round	Irreducible function g	$g(b, c, d)$
1	$F(b, c, d)$	$(b \wedge c) \vee (\sim b \wedge c)$
2	$G(b, c, d)$	$(b \wedge d)(c \wedge \sim d)$
3	$H(b, c, d)$	$b \oplus c \oplus d$
4	$I(b, c, d)$	$c \oplus (b \vee \sim d)$

performed according to a predetermined order. $T[i]$ is the above-mentioned 16 elements consisting of integer parts of $0.703125(i) * 81349 \bmod 256$, which are values that minimize the duplication of the hash values. In addition, g is an irreducible function of the bit operations, as shown in Table 1. Once four rounds of the data processing steps are completed, a 32-bit output is returned as a hash value result.

Compared to the MD5 algorithm, the definition and operation structure of the irreducible function of TinyMD5 are almost the same. The only differences are the size of the operation data, the initialized MD buffer values, and $T[i]$ values. MD5 uses 512-bit operation, whereas TinyMD5 is optimized for 128-bit operation. For initialized MD buffer values, MD5 uses 0x67452301, 0xefcdab89, 0x98badcfe, and 0x10325476, whereas TinyMD5 uses 0x8a, 0xdb, 0x75, and 0x24 for 32-bit operation. These values were selected to minimize the duplication of the hash values during the 32-bit operation. For $[i]$ $(i = 1, \ldots 64)$ used to remove the duplication of the hash values, and MD5 uses $2^{32} * \text{abs}(\sin(i))$, whereas the proposed algorithm uses values of $0.703125(i) * 81349 \bmod 256$. These values are used to select 64 random values that are not duplicates and are expressed with 8 bits.

In this paper, we propose a secure scheme that transmits data energy efficiently when we collect environmental data such as temperature and humidity through wireless sensor networks with limitations of data processing, storage, and communications. The proposed scheme encrypts original data by using values generated through the typically used MD5. That is, the proposed scheme generates a 128-bit hash value in the same way as MD5. However, the proposed scheme generates a hash value with 128 bits as an original data; the proposed scheme creates a reduced 32-bit hash value from the 128-bit hash value through the block data processing. As a result, MD5 generates a 128-bit hash value from the original data, while TinyMD5 generates a 32-bit hash value that reduces the 128-bit hash value. The original data can be recovered from data that has undergone the block data processing step by adding and using range information during the data recovery. The range information is expressed in 4 bits for every attribute value, while the range information result is converted into hexadecimal numbers to prevent the identification of the information even if it is added to a hash value. The next example describes the process of converting data using the proposed scheme and adding range information. Table 2 shows an example of the basic data structure that is utilized in a general sensor network. When a sensor node collects data, as shown in Table 2, the data is expressed as follows: [Sen_0_2, 60, 101, 16.51, 83.99, 28814, 2731, 191, true]. The hash value produced after the block data processing step is [d73e8a8c].

To reduce the complexity of decryption, the range information for each attribute is added. The following shows the operation process of range information calculation:

```
for i = 0 to i < 4
    if x < center
        RangeInfo_i = 0
    else
        RangeInfo_i = 1
    AND
    if x = 0
        Center = Max
    else
        Center = Min
end for
```

Here, Max, Min, center, x, and RangeInfo represent the maximum value, minimum value, center value, corresponding value, and range information, respectively. Using the sensing range of each attribute, the center value is calculated using (1) based on the Max and Min of the attribute values. If a sensing value is larger than the center value, it has a value of 0. If a sensing value is smaller than the center value, it has a value of 1. The range information is represented in 4 bits for every attribute value:

$$C_{\text{center}} = \frac{\text{Min} + \text{Max}}{2}. \quad (1)$$

Figure 5 shows the range information creation process. When the temperature is 16.51°C, the sensing range of

Table 2: Examples of data collected in a sensor node.

ID	X	Y	Temp	Humidity	Illuminance	CO_2	O_3	Operation
Sen_0_2	60	101	16.51	83.99	28814	2731	191	True

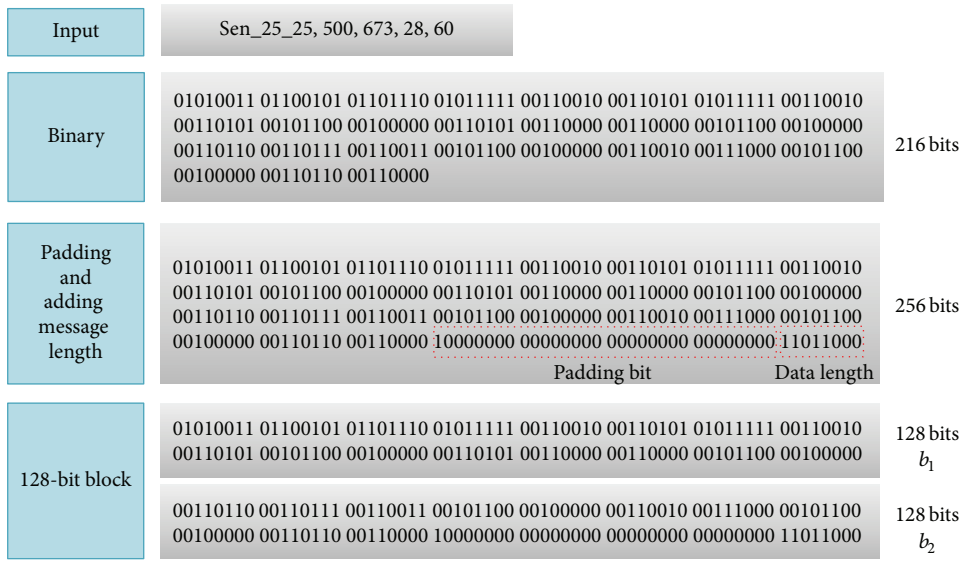

Figure 3: Padding bit additions and data length insertions.

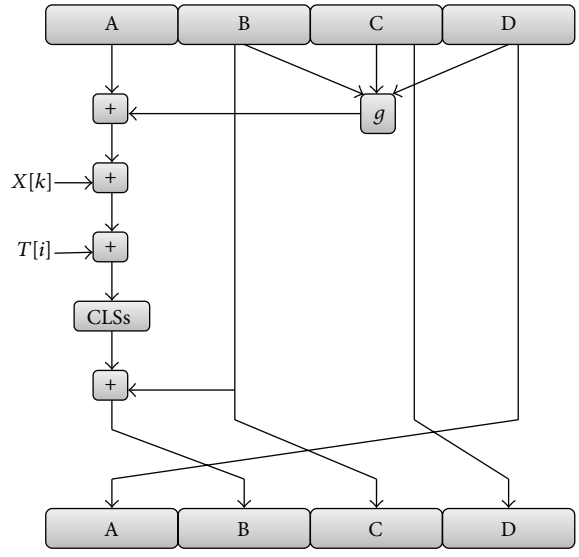

Figure 4: Operation structure of data processing step.

the temperature sensor is −40°C~125°C. The center value calculated using this value is 42.5°C, and 16.51°C is smaller than the center value, so the first range information has a value of 0. Since the first range information has a value of 0, the center value becomes the Max. Therefore, the center value is calculated again to find 1.25°C, so that the second range information has a value of 1. Through the above process, the binary expression of the range information can be expressed as 0101, which is then expressed as 5, a hexadecimal number, so as not to distinguish from the hash value.

Through the above process, the range information value in Table 2 can be expressed as [5d7e1], which is finally added to the TinyMD5 conversion value. Figure 6 shows the overall TinyMD5 processing steps using the above-explained example. If the original data is [Sen_0_2, 60, 101, 16.51, 83.99, 28814, 2731, 191, true], then the hash value after the block data processing step is [d73e8a8c], while the range information is [5d7e1]. Therefore, the value result of TinyMD5 is [d73e8a8c5d7e1].

3.3. Distributed Multipath Data Transmission. Since original data can be hacked if the full data of the packets is lost after the original data has been converted using TinyMD5, the converted data is divided and transmitted through multiple paths in a distributed manner. The distributed multipath data transmission step is performed through data division and processing. A preprocessing process is conducted with converted data for distributed data transmission after the data conversion process is finished using TinyMD5. A data value converted via TinyMD5 is divided into two parts. Divided converted data are transmitted to a base station and then recovered. Here, the ordering information of divided data is required to recover data. Therefore, the ordering information and a parity bit, which is required to detect errors, are added to the transferred data. Table 3 shows a data packet structure after data division and processing are completed.

Figure 7 shows the data division and processing procedures schematically using the above example for TinyMD5. As shown in Figure 7, if the original data is [Sen_0_2, 60, 101, 16.51, 83.99, 28814, 2731, 191, true], then the hash value after the block data processing step is [d73e8a8c], while

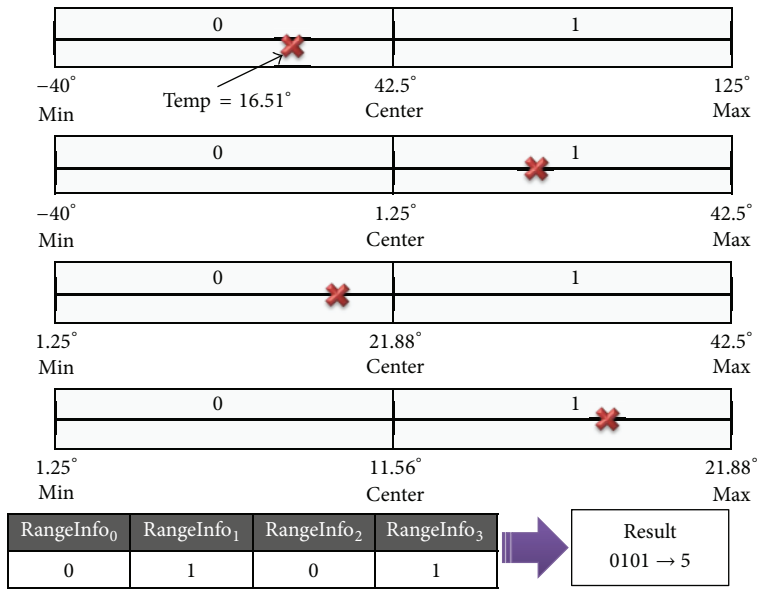

Figure 5: Example of range information creation.

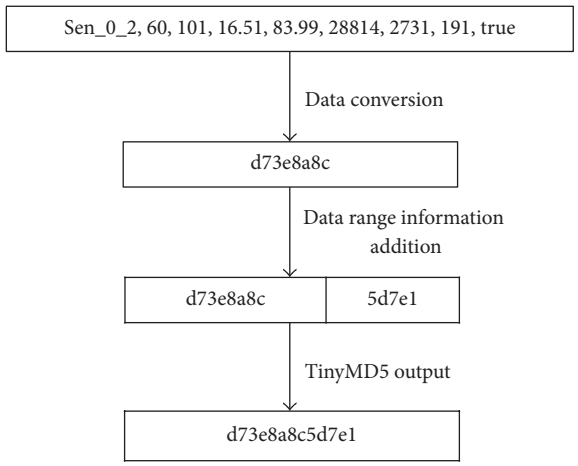

Figure 6: Example of TinyMD5 execution process.

Table 3: Transferred data structure after completing preprocessing process.

Ordering information	Converted value of divided MD5	Parity bit

the range information is [5d7e1]. Therefore, the value result of TinyMD5 is [d73e8a8c5d7e1]. Here, the result value is divided into two parts, which are [d73e8a8] and [c5d7e1]. The ordering information and a parity bit are added to each piece of the divided data.

After the data division and processing procedures are completed, the data is transmitted to a base station through multiple paths. The concept of an intermediate node is employed to transfer data through different paths. If divided data are transmitted via the same path and malicious nodes are arranged in neighboring paths, it is vulnerable to multiple data exposures. Thus, full data exposure is prevented by using distributed random data transmission. Divided data are passed through different intermediate nodes to ensure that data are transmitted through multiple paths to a base station. Here, it is assumed that every sensor node identifies its coordinates and base station.

A sensor computes its center coordinate and base station center coordinate using the coordinate values. On the basis of the computed coordinate values, a set of intermediate node candidates is created within a distance to the center coordinate of less than a certain range. Here, the values of α and β are the two range setup values for the set of intermediate node candidates, which are arbitrary values determined by considering network size or security. In the candidate set, two intermediate nodes are selected. The method for selecting intermediate nodes is shown in (2). Here, x_c, y_c, x_b, y_b, x_s, y_s, x_i, and y_i are x coordinate of center, y coordinate of center, x coordinate of base station, y coordinate of base station, x coordinate of sensing node, y coordinate of sensing node, x coordinate of intermediate sensor, and y coordinate of intermediate sensor:

$$\begin{aligned} (x_c, y_c) &= \left(\frac{x_b + x_s}{2}, \frac{y_b + y_s}{2}\right), \\ x_c - \alpha \le x_i \le x_c + \alpha, \\ y_c - \alpha \le y_i \le y_c + \alpha. \end{aligned} \tag{2}$$

The two coordinates calculated by (2) are set up as intermediate coordinates. Figure 8 shows the data transmission process through multiple paths. Each piece of divided data processed at a source node is transmitted using the intermediate coordinate information. The node that is closest to the intermediate coordinate is then set to an intermediate node, from which data are transmitted using the location information of a base station (sink node).

ID	X	Y	Temp	Humidity	Illuminance	CO_2	O_3	Operation
Sen_0_2	60	101	16.51	83.99	28814	2731	191	True

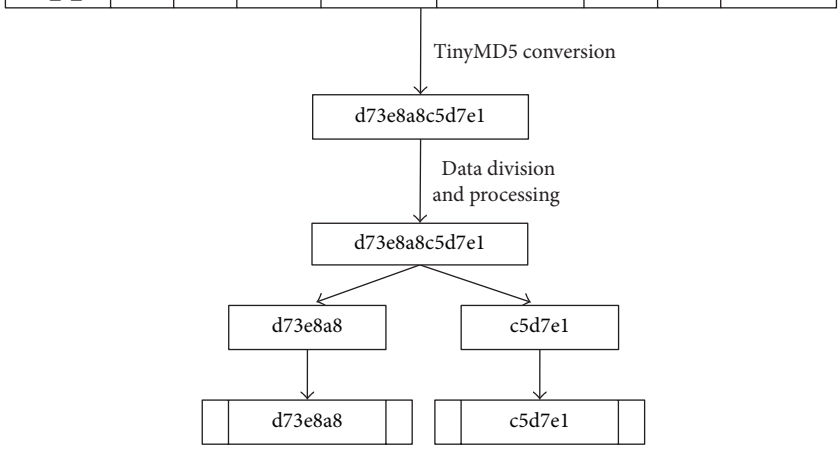

Figure 7: Example of data division and processing procedures.

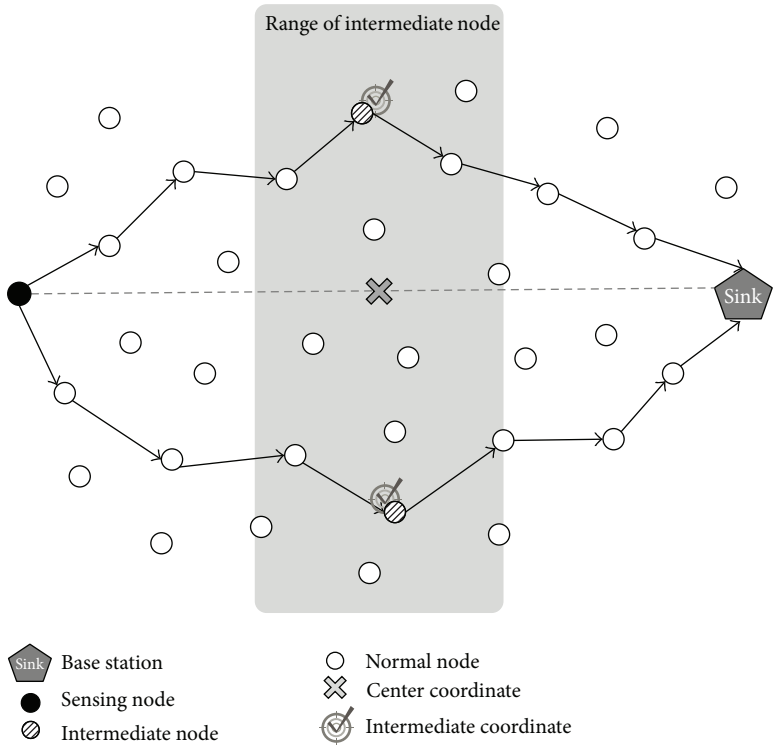

Figure 8: Distributed two-division data transmission.

3.4. Data Recovery. The original data are recovered from the data transferred to the base station through the data recovery process. The data recovery process consists of two subprocesses such as a matching decryption process that finds the original data and a filtering process that solves the collision problem of hash value that occurs due to a short hash value. Since a one-way hash function does not have a decryption algorithm, the data are recovered through data matching. However, during the data matching process, brute-force matching is required for the large number of TinyMD5 hash values stored in the database and the TinyMD5 hash values of original data, which generate an excessive computational load. To reduce this load, only data in a certain range are compared using the range information of each attribute to minimize the number of cases for data comparison, thereby reducing the required matching computation.

Divided data received at the base station are arranged and merged based on ordering information regardless of the receiving order. Then, only range information is extracted to calculate a data range that includes original data using the range information of each attribute. After this, a matching decryption process is conducted using the hash value results that do not include range information to recover the original

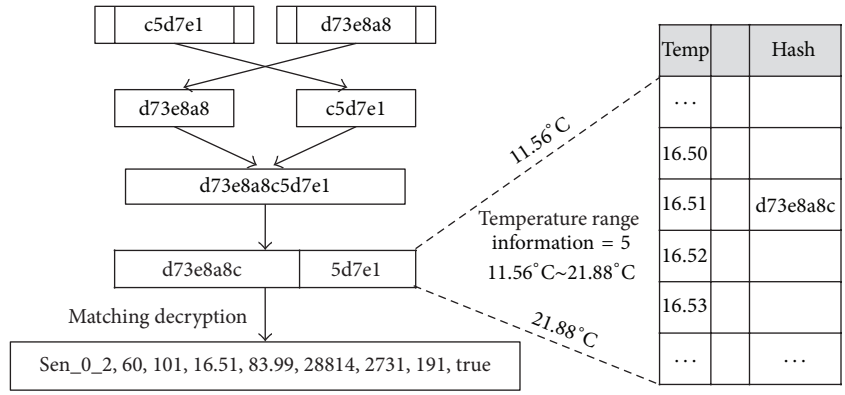

Figure 9: Matching decryption process.

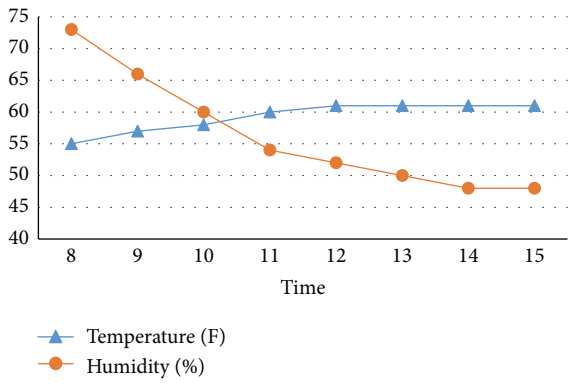

Figure 10: Temperature and humidity of San Francisco.

data. Figure 9 shows an example of the data decryption process. If the transferred range information is [5d7e1] and 5 indicates the range information of the temperature, the temperature value of the transferred data will be within a range of 11.56°C~21.88°C via the maximum and minimum values of each attribute. In addition to the temperature, the hash values of the original data are calculated using the range information of other attributes.

In general, wireless sensor networks have data reliability problems due to the characteristics of wireless transmission. The proposed scheme is to provide a security scheme that can be used in the infrastructure services layer. Data transmission errors in wireless sensor networks occur in the transport layer and the network layer. Many studies for guaranteeing data transmission reliability have been done in the transport and network layers. Therefore, the proposed scheme overcomes the data transmission errors in the transport and network layers by using the existing schemes [24, 25]. It also is possible to estimate data through data recovery when all of the packets are not lost. Typical example data collected in sensor networks are environmental data such as temperature and humidity. Figure 10 shows the temperature and humidity of San Francisco [26]. The data values collected in wireless sensor networks change continuously as shown in Figure 10. Therefore, we can estimate the lost data through the estimation of a data occurrence range. Most of the data collected in a sensor network is environmental data, such as temperature or humidity. Such environmental data is normally characterized by numerical physical continuity unless abnormal circumstances exist. Since the hash value is reduced to a 32-bit output, a hash value collision will occur accordingly. It is necessary to minimize hash value collisions with the following method. All data that corresponds to collided hash values is extracted, and data with an environmental value that is not similar to that of the source's neighboring nodes is removed based on the above assumption that environmental data such as temperature or humidity should have physical continuity.

4. Performance Evaluation

To show the superiority of the proposed scheme, various performance evaluations were conducted. Two factors were evaluated for performance comparison. First, the performance evaluation of hash value duplications was conducted by changing attribute values variably to validate the fewer hash value collisions of the proposed algorithm. Second, the performance evaluations of energy consumption and network lifetimes were conducted to demonstrate the energy efficiency of the proposed scheme. In the sensor network, each sensor node performs periodic data transmission or event based data transmission. This performance evaluation is based on periodic data transmission that each sensor node transmits its monitoring data to a base station periodically. The performance evaluations were implemented in Eclipse using Java programing language in a Windows 7 Operating System environment. Table 4 shows the environment where performance evaluations were conducted to analyze hash duplications. A sensor module generated the values of temperature, humidity, and illuminance randomly to produce approximately 30 million hash values to evaluate hash value duplications.

To evaluate the energy efficiency of the proposed scheme, the existing and proposed schemes were compared in terms of their energy consumption and network lifetimes. Table 5 shows the environment used for the experiment. The wireless sensor network is 1,000 m × 1,000 m. In the wireless sensor network, 2,500 nodes were deployed randomly, while a base

Table 4: Environment for performance evaluation of hash value duplication.

Parameter	Value
Temperature (emp)	−40~125
Humidity (%)	0~100
Illuminance (Lux)	0~100
Number of hash data	33,532,000

Table 5: Environment for performance evaluation of energy consumption.

Parameter	Value
Size of sensor network (m × m)	1,000 × 1,000
Number of sensors	2,500
Sensor communication radius (m)	35
Size of sensing data (bytes)	4
Initial energy of each sensor (J)	25

station was located at the middle of the right side of the network. It was assumed that all sensor nodes could communicate with neighboring nodes within a communication radius with lossless communication. The size of transmission data in a sensor node was assumed to be 4 bytes, while the communication radius and initial energy of each sensor were set to 35 m and 25 J, respectively.

Equations (3) to (7) were used as energy consumption models in this paper. R_{cost} in (3) refers to the consumed energy while receiving data, while T_{cost} in (4) refers to the consumed energy while transferring data. These two values have the same energy consumption. T_{amp} in (5) refers to consumed energy while amplifying a signal depending on the distance to a receiving node during the data transmission. RE_{cost} in (6) is a receiving cost, which is calculated by multiplying the message size by the energy consumed while receiving data. TR_{cost} in (7) refers to the transmission cost, which is calculated by multiplying the message size by summed value of the total value of energy consumed during the data transmission and the square of the distance between the receiving and transferring nodes multiplying by the consumed energy while amplifying a signal:

$$R_{\text{cost}} = 50 \times 10^{-9}\ [\text{J}], \tag{3}$$

$$T_{\text{cost}} = 50 \times 10^{-9}\ [\text{J}], \tag{4}$$

$$T_{\text{amp}} = 0.0013 \times 10^{-12}\ [\text{J}], \tag{5}$$

$$\text{RE}_{\text{cost}} = \text{MSG}_{\text{size}} \times R_{\text{cost}}, \tag{6}$$

$$\text{TR}_{\text{cost}} = \text{MSG}_{\text{size}} \times \left(T_{\text{cost}} + T_{\text{amp}} + T_{\text{dist}}^2\right). \tag{7}$$

Sensing data are collected in a sink node according to a certain period or requirement. Every sensor node has limited energy, and each sensor node can have a different battery capacity. Every sensor node has a wireless transmitter and receiver that can adjust a signal range. In addition, many pieces of data can be aggregated or merged into a single data size.

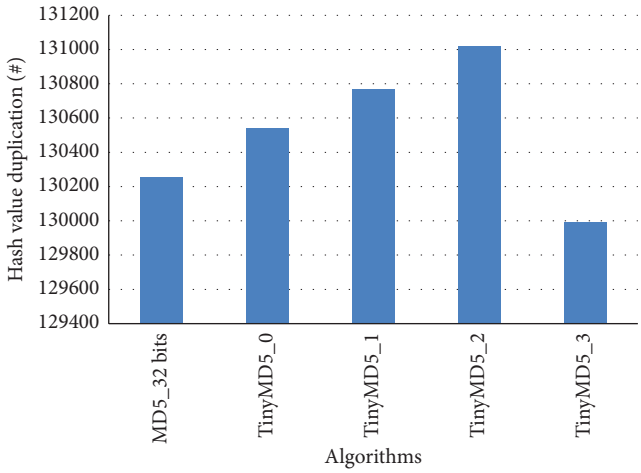

Figure 11: Hash value duplications according to algorithm execution.

Figure 11 shows the hash value duplications according to the algorithms. The performance evaluation was conducted based on the hash data results, in which the range information was not added. Data transmission errors in wireless sensor networks occur in the transport layer and the network layer. We assume that the proposed scheme overcomes the data transmission errors in the transport and network layers by using the existing schemes [24, 25]. In addition, since the data values collected in wireless sensor networks change continuously, we can estimate the lost data through the estimation of a data occurrence range. The proposed scheme encrypts original data through the typically used MD5. MD5 generates a hash value with 128 bits as an original data, while the proposed scheme creates a 32-bit hash value through the block data processing. The MD5_32 bits refers to one that simply reduces a hash value length from 128 bits to 32 bits. The output of the MD5_32 bits is the first eight of 32 characters, which are expressed as a hexadecimal number of MD5. TinyMD5_0 uses 0x6745, 0x2301, 0x98ba, and 0xdcfe as its buffer initialization values, while $2^{32} * \text{abs}(\sin(i))$, an equation that calculates $T[i]$, is used as the 216-operation value. The TinyMD5_1 processes padding bits use 112 mod 128, while the 128-bit block unit data processing buffer initialization values are set to 0x89, 0xab, 0xcd, and 0xef. $T[i]$ uses values calculated via the $0.703125(i) * 81349 \bmod 256$ used rather than $2^{32} * \text{abs}(\sin(i))$, while a circular bit value is calculated from 8 bits. TinyMD5_2 is the same as TinyMD5_1 except for the initialization buffer values, which are 0x8a, 0xdb, 0x75, and 0x24. TinyMD5_3 is the same as TinyMD5_2 except for the bit circulation, which is 32 bits rather than 8 bits. Based on the performance evaluation results, TinyMD5_3 was adopted as the proposed TinyMD5 algorithm since it generated the fewest hash value duplications.

In the performance evaluation, the proposed scheme was compared with DES [17], AES [18], and MD5 [15], which are widely utilized encryption schemes in the existing wireless sensor networks. Here, TinyMD5 only transmitted hash data in which range information was not added. Data transmission utilized GPSR [16] and a two-division GPSR transmission scheme, while the performance evaluation considered energy

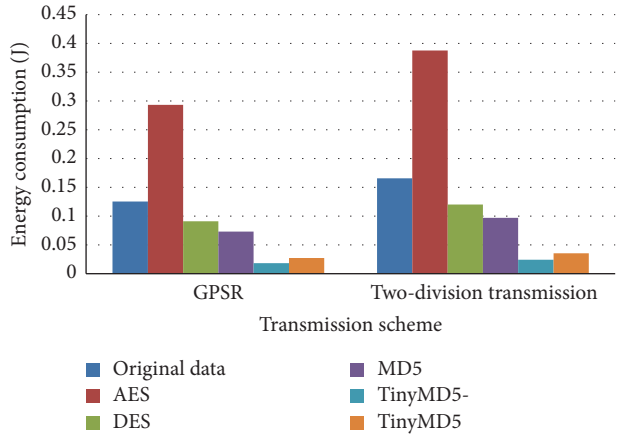

Figure 12: Energy consumption of encryption algorithms according to data transmission schemes.

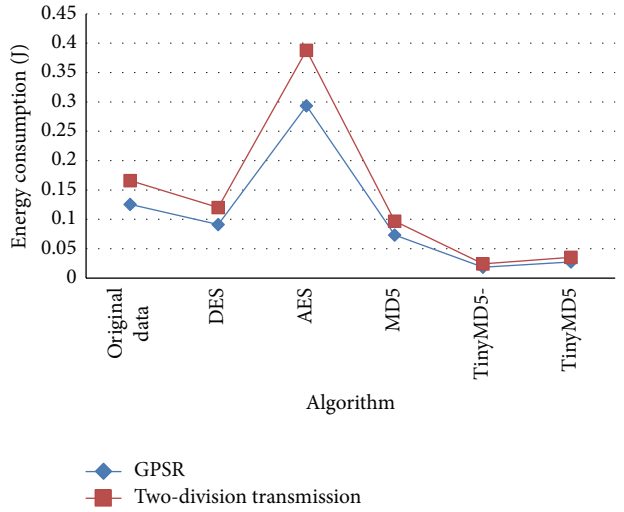

Figure 13: Energy consumption differences of algorithms according to transmission schemes.

consumption in the overall sensor network. Figure 12 shows the energy consumption in encryption schemes according to the data transmission schemes. During the data encryption process, DES encryption, AES encryption, and MD5 generated 160-bit, 512-bit, and 128-bit data, respectively, while TinyMD5- and TinyMD5 generated 32-bit and 52-bit data, which were relatively low transmission amounts compared to existing encryption schemes. The performance evaluation results showed that the proposed algorithm showed energy consumption reductions of 39%, 12%, and 37% over DES and AES encryption and MD5 and an average reduction of 29% compared to the existing schemes.

Figure 13 shows the energy consumption differences of algorithms according to the data transmission schemes. Since the two-division data transmission scheme transfers data as it bypasses paths, it consumes more energy than the GPSR transmission scheme. However, it was verified that the proposed scheme that uses the TinyMD5 algorithm had the smallest difference in energy consumption between GPSR

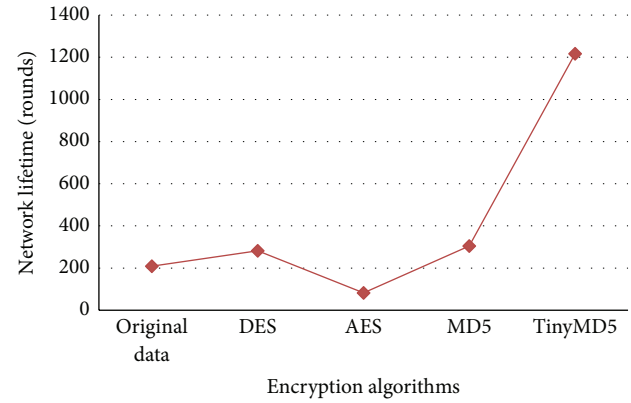

Figure 14: Sensor network lifetime according to encryption algorithms.

and two-division transmission schemes compared to existing algorithms. In addition, the two-division data transmission in the proposed scheme consumed less energy compared to using only the GPSR scheme with existing algorithms while transferring data. Such characteristics can resolve issues of excessive energy consumption due to security measures in other security schemes.

Figure 14 shows the comparison results of sensor network lifetimes according to encryption algorithms. The performance evaluation of the sensor network lifetime was conducted based on the assumption that 30% of the sensor nodes consumed all the energy. The performance evaluation results showed that the TinyMD5 algorithm in the proposed scheme improved the network lifetime by approximately seven times, as it required relatively less communication than other schemes as a result of shorter transferred data lengths.

5. Conclusion

In this paper, we proposed an energy-efficient securing scheme for wireless sensor networks to improve energy efficiency as well as security while minimizing energy consumption by restricting the decryption of exposed data even if transmitted data are intercepted in a wireless sensor network. The proposed scheme used a TinyMD5 algorithm, a variant of MD5, which is a one-way hash function, to suit the characteristics of wireless sensor networks. Moreover, the number of communication messages among nodes was minimized by transmitting divided data with GPSR, thereby using energy efficiently while strengthening security. The proposed scheme has the advantage of having no communication influence in the sensors despite data size increases, since the size of transmitted data is constant regardless of the data sizes used due to the use of the hash function. Through the performance evaluation, the proposed scheme showed reductions in communication costs and improvements in network lifetimes compared to the existing schemes, while security was maintained. In the near future, we will apply the proposed scheme to the real wireless sensor networks.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) Support Programs IITP-2016-H8501-16-1013 and IITP-2016-H8601-16-1008 supervised by the IITP (Institute for Information & Communication Technology Promotion), by Basic Science Research Program through the Korean National Research Foundation (NRF) funded by the Ministry of Education (2015R1D1A3A01015962), and by the ICT R&D program of MSIP/IITP [B0101-15-0266, Development of High Performance Visual BigData Discovery Platform for Large-Scale Realtime Data Analysis].

References

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

[2] J. Park, H. Park, D.-O. Seong, and J. Yoo, "A sensor positioning scheme with high accuracy in nonuniform wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 507605, 7 pages, 2013.

[3] T. M. Cao, B. Bellata, and M. Oliver, "Design of a generic management system for wireless sensor networks," *Ad Hoc Networks*, vol. 20, pp. 16–35, 2014.

[4] K. Sohrabi, J. Gao, V. Ailawadhi, and G. J. Pottie, "Protocols for self-organization of a wireless sensor network," *IEEE Personal Communications*, vol. 7, no. 5, pp. 16–27, 2002.

[5] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–105, 2002.

[6] C. Zhu, L. Shu, T. Hara, L. Wang, S. Nishio, and L. T. Yang, "A survey on communication and data management issues in mobile sensor networks," *Wireless Communications and Mobile Computing*, vol. 14, no. 1, pp. 19–36, 2014.

[7] X.-H. Li and Z.-H. Guan, "Energy-aware routing in wireless sensor networks using local betweenness centrality," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 307038, 9 pages, 2013.

[8] Z. Sun and Z. Shao, "Greedy forwarding routing strategy based on energy efficiency in wireless sensor network," *Journal of Networks*, vol. 8, no. 10, pp. 2317–2323, 2013.

[9] J. M. Kahn, R. H. Katz, and K. S. J. Pister, "Next century challenges: mobile networking for 'Smart Dust'," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 271–278, Seattle, Wash, USA, August 1999.

[10] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[11] J. Duan, D. Yang, H. Zhu, S. Zhang, and J. Zhao, "TSRF: a trust-aware secure routing framework in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2014, Article ID 209436, 14 pages, 2014.

[12] J. Zhou, "Efficient and secure routing protocol based on encryption and authentication for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, Article ID 108968, 17 pages, 2013.

[13] R. Lu, X. Lin, H. Zhu, and X. Shen, "TESP^2: timed efficient source privacy preservation scheme for wireless sensor networks," in *Proceedings of the IEEE International Conference on Communications (ICC '10)*, pp. 105–110, IEEE, Cape Town, South Africa, May 2010.

[14] P. Samundiswary, D. Sathian, and P. Dananjayan, "Secured greedy perimeter stateless routing for wireless sensor networks," *International Journal of Ad Hoc, Sensor & Ubiquitous Computing*, vol. 1, no. 2, pp. 9–20, 2010.

[15] R. Rivest, *The MD5 Message-Digest Algorithm*, MIT Laboratory for Computer Science and RSA Data Security Inc, 1992.

[16] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MOBICOM '00)*, pp. 243–254, August 2000.

[17] National Bureau of Standards (NBS), *Data Encryption Standard*, vol. 46 of *FIPS Publications*, National Bureau of Standards (NBS), 1977.

[18] NIST, "Advanced encryption standard," FIPS Pub. 197, U.S. Department of Commerce, 2001.

[19] http://en.wikipedia.org/wiki/Data_Encryption_Standard.

[20] http://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

[21] http://en.wikipedia.org/wiki/MD5.

[22] C. Karlof, N. Sastry, and D. Wagner, "TinySec: a link layer security architecture for wireless sensor networks," in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems (SenSys '04)*, pp. 162–175, Baltimore, Md, USA, November 2004.

[23] Y. Li and J. Ren, "Source-location privacy through dynamic routing in wireless sensor networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '10)*, IEEE, San Diego, Calif, USA, March 2010.

[24] Y.-R. Chuang, H.-W. Tseng, S.-T. Sheu, and C.-W. Su, "An almost overhead-free error control scheme for IEEE 802.16-based multi-hop networks," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '09)*, pp. 1–6, Honolulu, Hawaii, USA, December 2009.

[25] Y.-R. Chuang, H.-W. Tseng, and S.-T. Sheu, "A performance study of discrete-error-checking scheme (DECS) with the optimal division locations for IEEE 802.16-based multihop networks," *IEEE Transactions on Computers*, vol. 62, no. 12, pp. 2354–2365, 2013.

[26] http://www.accuweather.com.