*Article*

# A Randomized Watermarking Technique for Detecting Malicious Data Injection Attacks in Heterogeneous Wireless Sensor Networks for Internet of Things Applications

**Arwa Alromih [1],*, Mznah Al-Rodhaan [2] and Yuan Tian [2]**

[1] Information Systems Department, King Saud University, Riyadh 12371, Saudi Arabia; aalromih@ksu.edu.sa

[2] Computer Science Department, King Saud University, Riyadh 12371, Saudi Arabia; rodhaan@ksu.edu.sa (M.A.-R); ytian@ksu.edu.sa (Y.T.)

* Correspondence: aalromih@ksu.edu.sa; Tel.: +966-11-805-1241

**Abstract:** Using Internet of Things (IoT) applications has been a growing trend in the last few years. They have been deployed in several areas of life including secure and sensitive sectors like military and health. In these sectors, sensory data is the main factor in any decision-making process. This introduces the need to ensure the integrity of data. Secure techniques are needed to detect any data injection attempt before catastrophic effects happen. Sensors have limited computational and power resources. This limitation creates a challenge to design a security mechanism that is both secure and energy-efficient. This work presents a Randomized Watermarking Filtering Scheme (RWFS) for IoT applications that provides en-route filtering to get rid of any injected data at an early stage of the communication. Filtering injected data is based on a watermark that is generated from the original data and embedded directly in random places throughout the packet's payload. The scheme uses homomorphic encryption techniques to conceal the report's measurement from any adversary. The advantage of homomorphic encryption is that it allows the data to be aggregated and, thus, decreases the packet's size. The results of our proposed scheme proved that it improves the security and energy consumption of the system as it mitigates some of the limitations in the existing works.

**Keywords:** Internet of Things (IoT); wireless sensor network (WSN); data integrity; watermark; data injection attack

## 1. Introduction

The Internet of Things (IoT) is an ecosystem that interconnect "smart" objects that are connected to the internet. These objects sense data from the environment and route it to the internet to be collected, processed an analyzed using high technologies [1]. Wireless Sensor Networks (WSN) is one of the main enabling technologies for the Internet of Things [2]. They are composed of low-powered sensor nodes and a powerful base station. Sensor nodes have very limited computational and storing capabilities. Wireless sensor networks are used in a wide range of applications such as environmental monitoring and control, health care systems, office automation and home control [3]. In these applications, sensors that make up the networked system will share their data through the shared wireless medium which is vulnerable to eavesdropping attacks. Due to this reason and the limitation detailed earlier, they can be vulnerable to multiple numbers of security attacks that can affect the integrity of the data, such as packet drop, false data injection, data modification and packet replay [4]. It is very crucial to have a countermeasure to filter false data as early and accurately as possible. By injecting false data into the network, the attacker can make fake reports that will lead to serious damage or dangerous effects.

Some mechanisms have been developed to secure IoT of such attacks. These solutions vary regarding techniques. Some of these solutions use software attestation, anomaly detection techniques, trust-based techniques or signature-based techniques. However, those mechanisms have their limitations and cannot be used effectively to provide both confidentiality and integrity. Software attestation techniques require special hardware. Anomaly detection and trust-based techniques can suffer from a high rate of false positives [5]. They also involve extra steps of building a distribution of the normal behavior first. Signature-based techniques can be considered computationally complex [6]. The above weaknesses of these techniques have led to explore other methods.

To ensure the integrity of the data and efficient use of bandwidth, watermarking techniques are used in many applications. They have been originally used to protect multimedia content and relational databases [7]. Many researchers have used it in securing sensory data. In the watermarking process, every data is embedded with a unique watermark by the sensor node before sending it. The access point can then verify its integrity.

Commonly used data protection and integrity methods are complex [8]. Therefore, research effort should be made to design a new low-complexity scheme, yet secure data integrity system that can detect and filter out false data injection attacks. Although literature have solutions to provide filtering mechanisms, they are expensive, and some are an application or context-dependent [5].

The main contribution of this paper is the offer of a Randomized Watermarking Filtering Scheme (RWFS) for WSN, which can filter false injected data effectively and as early as possible without relying on any static node location or secure routing protocols. The proposed system aims to protect the integrity, authenticity, and confidentiality of sensory data using a lightweight watermarking technique. It will provide en-route filtering rather than end-to-end to minimize communication overhead by reducing the number of false data reports. Integrity and authenticity will be provided by embedding a watermark at random locations within the data. Embedding a watermark within multiple random locations will make it more difficult for an adversary to inject false report. The system will also use homomorphic symmetric encryption to provide confidentiality.

The main purpose of this paper is to address the following issues:

- Investigate a new technique for filtering malicious data injection attacks based on watermarking.
- Develop a novel energy-efficient scheme that aims to minimize both packet size and number of communications between nodes.
- Design a new and random way of embedding the watermark that will be based on PRNG.
- Develop a filtering model to evaluate the performance of the proposed watermark scheme and validate the analytic model with simulations.

The rest of this paper is organized as follows. Section 2 reviews related work presented by other scholars in different approaches. We address the different system models and assumptions in Section 3, and we present the proposed scheme in detail in Section 4. Section 5 presents a security analysis of the proposed scheme. Section 6 describes the performance evaluation methodology and results from the scheme simulation. A summary of the contributions of this work and a discussion of possible future work are presented in Section 7.

## 2. Literature Review

Many recent types of research have provided countermeasures against data injection attacks in IoT. These countermeasures can be divided into five main categories, using software attestation, anomaly detection, trust management, signature-based and finally digital watermarking techniques. These scholars have employed solutions to counter injection attacks of the sensory data of IoT applications.

Signature-based detection techniques have been the basis of much more research. In Dynamic En-route Filtering scheme (DEF) scheme [9], message authentication code (MAC) is used as a signature to verify the integrity of a report. Each sensing node will generate multiple authentication keys using one-way hash chains. These keys will be disseminated by the cluster head to the forwarding nodes using preloaded secret keys. DEF uses many keys since it involves using both authentication keys and secret keys. It also involves a vast number of communication messages between nodes to complete a single round.

In [10] and [11], a keyed-hash message authentication code (HMAC) is used to provide integrity. Cui, et al. in [12] proposed a new scheme to provide integrity and confidentiality. The authors have used a homomorphic MAC (H-MAC) algorithm to achieve end-to-end data integrity. The homomorphic MAC can be added i.e. $H-MAC(a+b) = H-MAC(a) + H-MAC(b)$. H-MAC is designed to check the integrity of aggregated data. Moreover, a homomorphic encryption algorithm is used in this scheme to provide confidentiality.

PCREF that was proposed in [13] uses a message authentication polynomial (MAP) to provide integrity. Every node is assigned a polynomial primitive used for authentication and checking. After a node sensed some data, it will generate a message authentication polynomial (MAP) and append to the measurement to be forwarded to the base station. The scheme only provides end-to-end integrity as it evaluates the integrity at the BS.

Another type of integrity mechanism is using watermarking techniques. There are three types of digital watermarks, robust, fragile, and semi-fragile watermarks. The first two types have been the subject of research in wireless sensor networks. Semi-fragile watermarking is not typically used in WSN. Work of using watermarking techniques for integrity and authentication in WSNs had begun from 2003. Feng et al. [14] proposed the first watermarking system that embeds cryptographically encoded signatures into data.

Fragile watermarking is used to provide data integrity authentication through embedding watermark into the original data. In [15], the authors have developed a hop-by-hop fragile watermarking technique. They have assumed the network to be homogeneous and formed as a tree with each cluster head to have at most three children. The technique uses data aggregation to reduce communication overhead. The watermark size is only one bit, and the generation process is very simple by using the XOR operator. The embedding is done at the least significant bit of the data. The technique is simple but not secure. CLWDA [16] is another fragile watermarking mechanism that adopts two types of watermarks. The first is called fragile watermark (FW) that is used to validate data sent from the first level of the network. The high level of the network (aggregator nodes) uses a reinforced fragile watermark (RW) which is more secure as the watermark is encrypted using the asymmetric key. The mechanism adopts a different embedding approach than most of the proposed schemes. The embedding position of the watermark will be dynamic, and it will be calculated based on the wakeup interval used by each node. Although this makes it somewhat random, it can be easily detected as it will form a pattern.

A watermarking algorithm based on a one-way hash function was proposed in [17] . The watermark is computed for each byte of data and then grouped using XOR function. The final watermark is then divided between the bytes and embedded in predefined redundant spaces. The proposed algorithm is computationally simple.

Ahmed et al. [7] had proposed a zero-watermarking technique (ZWT) that generates a watermark from the properties of a sensed data. After encrypting the watermark using a shared secret key, the sensor embeds the encrypted watermark at the end of the sensed data before it is sent to the base station. ZWT was tested against other watermarking techniques and results showed that it does not introduce any computation overhead as the generation process is very simple.

A semi-blind watermarking technique that uses linear interpolation to embed the watermark into the data was proposed in [18]. The main advantage of the technique is that it does not introduce any extra bits for the watermark. However, it uses a fixed watermark for all nodes in the system which can be easily cracked.

Robust watermarking used to provide copyright protection was combined with the fragile watermark in [19]. The authors proposed a digital watermarking scheme based on multiple functions to protect images transmitted in WSN. The scheme embeds two types of watermarks, robust and fragile watermarks to provide copyright protection and integrity respectively. The robust watermark is generated from the stable features of the original picture. While image pixels are used to generate the fragile watermark.

Guan et al. [20] proposed another robust watermarking scheme for node authentication. The proposed scheme computes the watermark based on a randomly chosen data sampling time interval. The scheme is not robust in detecting false data insertion attacks as it does not consider the whole data in generating the watermark.

Reversible watermarking techniques were proposed in [21], [22] and [23]. Shi et al. in [21] proposed a reversible watermarking algorithm based on prediction-error expansion. The algorithm groups every two adjacent data together. The first one is used to generate the watermark while the other is used as a carrier for it. The drawback of this method is that verification is end-to-end since verification needs extra storage as it uses buffering.

Ding et al. [22] had proposed a similar end-to-end algorithm called (RDE), but instead, the watermark is generated based on the difference expansion. The algorithm groups $n$ data items and uses them all to calculate the watermark. The embedding process calculates the weighted difference between each data item with the first data. The embedding process is straightforward every bit of the watermark is embedded in the least significant bit of each data item. This algorithm can ensure the complete recovery of data at the base station.

In [23], the authors have proposed an improved version of different expansion used in [22]. The whole network was assumed to be viewed as an image with each sensor node as a pixel. The technique generates a single watermark for the whole system data and split it into several segments. Each of which is assigned to random sensors for embedding. The extracting operation will use a location map to know which sensor nodes were chosen to embed the watermark.

## 3. System Models and Assumptions

### 3.1. Network Model

The network will be assumed to consist of $N$ nodes that are uniformly randomly distributed within an $M \times M$ square area. Nodes are heterogeneous in terms of energy i.e., there are two types of nodes normal nodes and advanced nodes. Normal nodes have energy $E_0$ while advanced nodes have $a$ times more energy than normal nodes $E_0(1+a)$. This will mean that some nodes are equipped with more energy than other nodes. Sensor nodes are formed into multiple clusters. Clusters will each have $n$ sensing nodes and a cluster head $CH$. These two types of nodes are shown in Figure 1. Nodes that belong to the same cluster will know the IDs of each other and can communicate with each other. Routing the data to the base station is done through the CH nodes.

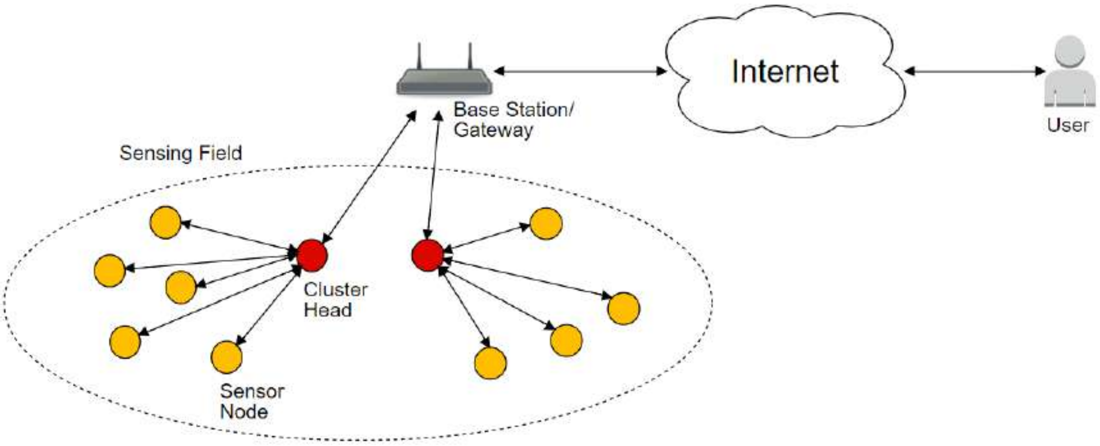


Figure 1: Network Model

### 3.2. Clustering Model

Clustering is used as a routing protocol in the network. Cluster-based routing is an efficient approach to maintain system scalability, lifetime, and energy efficiency within a cluster. It enables clusters to perform data aggregation in order to decrease the number of transmitted messages to the AP. In our schemed, clusters are formed using the Enhanced Distributed Energy Efficient Clustering (E-DEEC) protocol where cluster head selection is made based on nodes' residual energy [24]. E-DEEC uses a concept of the three-level heterogeneous network where the system has three types of nodes: normal, advanced and supernodes. Each type of node has its own initial energy. Normal nodes have $E_0$ energy. Advanced nodes have $a$ times more energy than normal nodes, i.e., $E_0(1+a)$. While, super nodes have $b$ times more energy than the normal ones, i.e., $E_0(1+b)$.

In E-DEEC, cluster heads are chosen based on a probability value $p_i$ that is computed differently based on the node's type. $p_i$ can be computed as follows:

$$p_i = \begin{cases} \frac{p_{out}E_i(r)}{(1+m(a+m_0 b))\bar{E}(r)} & if\ s_i\ is\ normal\ node \\ \frac{p_{out}(1+a)E_i(r)}{(1+m(a+m_0 b))\bar{E}(r)} & if\ s_i\ is\ advanced\ node, \\ \frac{p_{out}(1+b)E_i(r)}{(1+m(a+m_0 b))\bar{E}(r)} & if\ s_i\ is\ super\ node \end{cases} \tag{1}$$

### *3.3. Radio Energy Dissipation Model*

To analyze the performance of our proposed protocol, we need to know the amount of energy consumed by a sensor node. Therefore, we use similar Radio Energy Model as the one proposed in [25]. According to [25], the radio energy dissipation model is illustrated as Figure 2.

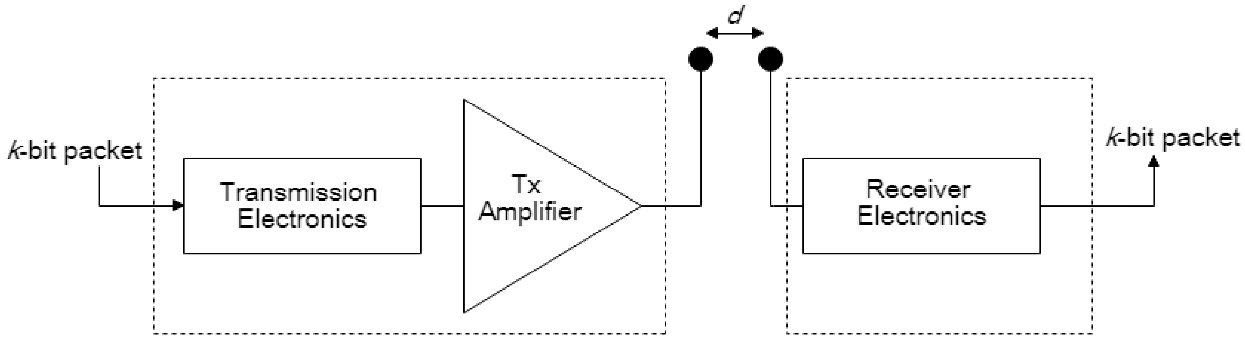


Figure 2: Radio Energy Model

In this model, both the free space ($d^2$ power loss) and the multipath fading ($d^4$ power loss) channel models are used. If the distance is less than a threshold *do*, the free space model is used; otherwise, the multipath model is used. Thus, to transmit $k$ bit packet for a distance $d$, the consumption energy is computed as [25]:

$$E_{Tx}(k,d) = \begin{cases} k \times E_{elec} + k \times E_{fs} \times d^2 & if\ d < d_0 \\ k \times E_{elec} + k \times E_{mp} \times d^4 & if\ d > d_0 \end{cases} \tag{1}$$

and to receive $k$ bit packet, consumption energy is [25]:

$$E_{Rx}(k) = k \times E_{elec} \tag{2}$$

where $E_{elec}$ denotes electronics energy, which depends on factors such as the digital coding, modulation, filtering and spreading of signal. $E_{fs}$ is the free space power loss and $E_{mp}$ is the multipath fading loss. The Value of threshold distance $do$ is given by $do = \sqrt{\frac{E_{fs}}{E_{mp}}}$ .

From equations (2) and (3), it is clearly shown that the energy consumed for data transmission increases as the packet size increase and square or biquadratic of distance, while energy consumed for reception is only proportional to the packet size [26].

### *3.4. Adversary Model*

In this paper, we consider an adversary model that is similar to the one used in [12, 27]. The model has three types of adversaries:

1. **A passive adversary** that can observe and obtain packets by eavesdropping the network transmission.
2. **An external adversary** who can originate and inject false data from the outside. He can also replay old packets or modify the transmitted data.
3. **An internal adversary** that can physically compromise sensor nodes or cluster heads.

Each one of the three types of adversaries can launch a different number of attacks based on his abilities mentioned earlier.

An adversary of **Type 1** can obtain secret information by eavesdropping.

a. **Eavesdropping Attack:** this attack concerns the passive adversary who aims to get information by listening to the message transmission in the broadcasting wireless medium [28].

The adversary of **Type 2** can send false information by modifying the content of packets, injecting false data ore replaying old messages. Therefore, he can launch three kinds of attacks:

a. **Injection Attack:** In this attack, an adversary injects false data into the network and compromises the trust of the information communicated.
b. **Replay Attack:** the attacker captures some packets and resends them in a future time.
c. **Modification Attack:** the adversary modifies the data without knowing the content.

Lastly, the adversary of **Type 3** can sense and know all secret parameters. In this situation, he can launch a compromise attack.

a. **Node Compromise Attack:** the attacker physically comprises a node or CH to obtain secret information and generate secret parameters of the network.

## 4. Proposed Scheme

The proposed system provides data integrity, authenticity and confidentiality by embedding a watermark at random locations within the data and using symmetric encryption. The proposed system consists of three phases: (1) *Setup and Key Management*. (2) *Sensing and Reporting*. (3) *Verifying and Aggregation*.

In the *Setup and Key Management* phase, the clusters are formed, and all authentication information is assigned to the sensors. After that in the case of an event or based on periodic period, sensor nodes start to sense information, encrypt it, embed the watermark at a random place and then send the final report $R_i$ to the cluster head to start the verifying step. In the *Verifying and Aggregation* phase, the cluster head checks whether the received report is valid or not to filter bogus packets. The verification is done by regenerating the watermark from the extracted data. For each valid $R_i$ of the received reports, *CH* aggregates the sensed data, embed a new watermark of the aggregated data and forward it to the access point. The AP individually decrypts and verifies the reports of each cluster as it is assumed that the AP has all the necessary keys and parameters.

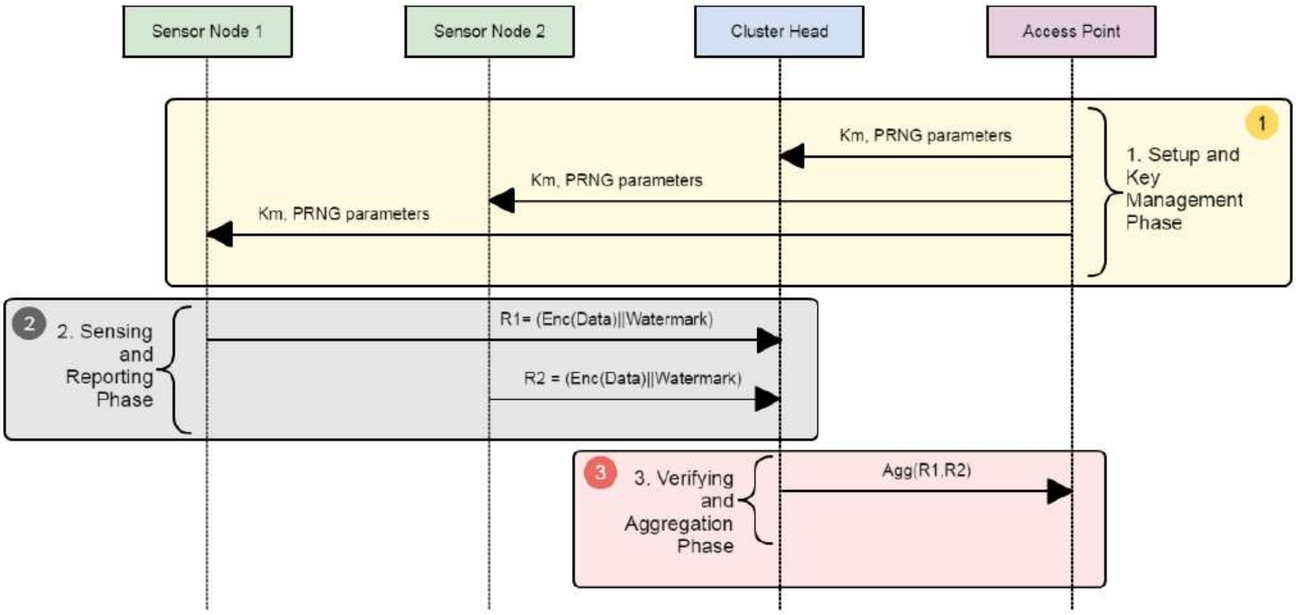

Sensor Node 1
Sensor Node 2
Cluster Head
Access Point
1
Km, PRNG parameters
Km, PRNG parameters
Km, PRNG parameters
1. Setup and Key Management Phase
2
2. Sensing and Reporting Phase
R1= (Enc(Data)||Watermark)
R2 = (Enc(Data)||Watermark)
3
3. Verifying and Aggregation Phase
Agg(R1,R2)

Figure 3: Sequence diagram of the proposed scheme

Following, the phases of RWFS are described in details and all notations used later are listed in Table 1.

Table 1: System notations and their description

| Notation | Description |
|---|---|
| $CH_j$ | The cluster head node of the *j*th cluster in the network |
| $S_i$ | The sensor node *i* |
| $n$ | Number of sensor nodes per cluster |
| $N$ | Number of sensor nodes in the whole network |
| $CH_jID$ | Cluster identifier of cluster $CH_j$ |
| $K_{cj}$ | The cluster key of the *j*th cluster in the network |
| $K_m$ | The master key of the network |
| $W_i$ | Watermark of the sensor node $S_i$ |
| $D_i$ | Sensing data of the sensor node $S_i$ |
| $E_i$ | Encrypted sensed data of the sensor node $S_i$ |
| $R_i$ | The complete report of the sensor node $S_i$ |

*Phase1: Setup and Key Management*

In the setup step, the network uses E-DEEC clustering algorithm to cluster the network into *j* clusters. In each cluster *j*, $n$ sensing nodes and a $CH$ will be assigned an $ID$. These nodes can communicate with the CH directly. After that, the AP initializes each node with the following $\{m_j, c_j, a_j, H(,), k_m\}$ where $K_m$ is the master key of the system that is used by each sensing node in the following function to produce the cluster key $K_c$

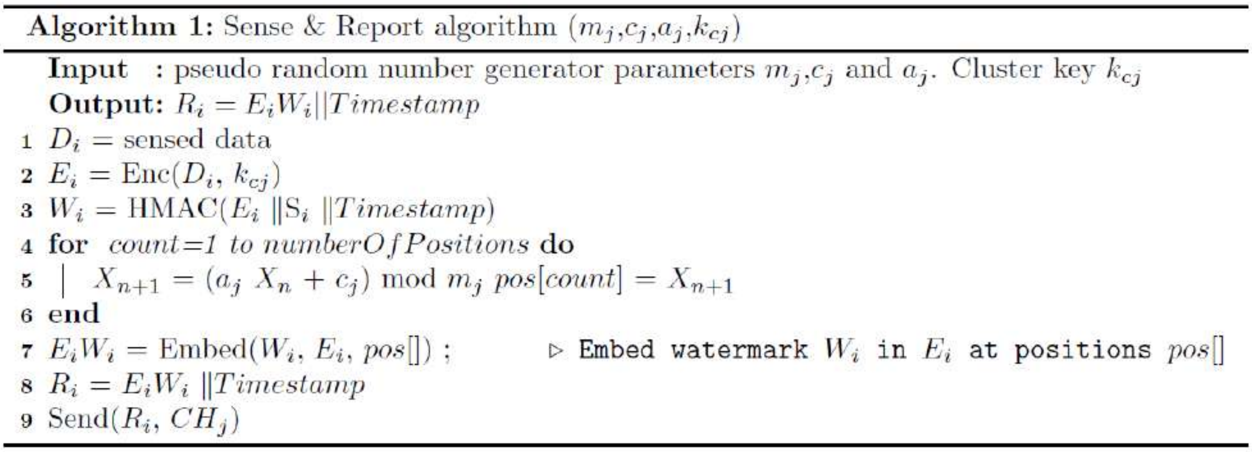
$$\boldsymbol{K_{cj} = k_{\,m} \oplus CH_jID \mid H\,(K_m)} \tag{4}$$

$m_j, c_j$ and $a_j$ are the parameters used for the pseudo-random number generator function. These parameters differ among clusters.

*Phase2: Sensing and Reporting*

Each sensing node $S_i$ measures the data that it monitors from the environment as $D_i$ and then encrypt it as the following: $E_i = Enc(D._iK_{cj})$. After that, the node generates the watermark of the encrypted measure as $W_i$. The embedding position of the watermark $W_i$ is randomly chosen and varies frequently. The position of each nibble (4 bits) of $W_i$ is generated using a PRNG. The PRNG is used to generate three random numbers for three different positions. Finally, after embedding the watermark, the report is sent as the following format to $CH_j$

$$R_i = (E_i \mid W_i \mid Time\,) \tag{5}$$



```
Algorithm 1: Sense & Report algorithm ($m_j$,$c_j$,$a_j$,$k_{cj}$)
  Input  : pseudo random number generator parameters $m_j$,$c_j$ and $a_j$. Cluster key $k_{cj}$
  Output: $R_i = E_iW_i||Timestamp$
  $D_i$ = sensed data
  $E_i$ = Enc($D_i$, $k_{cj}$)
  $W_i$ = HMAC($E_i$ ||S$_i$ ||$Timestamp$)
  for  $count$=1 to $numberOfPositions$ do
  |   $X_{n+1} = (a_j\ X_n + c_j) \bmod m_j\ pos[count] = X_{n+1}$
  end
  $E_iW_i$ = Embed($W_i$, $E_i$, $pos$[]) ;          ▷ Embed watermark $W_i$ in $E_i$ at positions $pos$[]
  $R_i = E_iW_i$ ||$Timestamp$
  Send($R_i$, $CH_j$)
```

Figure 4: Pseudo code for Sensing and Reporting phase

Encryption Algorithm

As mentioned in the introduction, the scheme uses a homomorphic encryption algorithm to encrypt data so it can be aggregated. Moreover, since encryption algorithms are usually expensive and complex to compute, we have used a very lightweight encryption algorithm that was proposed in [29]. The encryption function that was proposed by Castelluccia, Claude, et al. is as follows $c = Enc(d, k, M) = d + k(modM)$, where d is the data to encrypt, k is the secret key for the node and M is the modulus.

Watermarking Algorithm

The primary goal of the proposed watermarking algorithm is to randomly embed the watermark within the data but still can manage to extract it. Also, RWFS strives to resolve and effectively maintain the energy level of the nodes. The algorithm consists of two procedures: watermark generation and watermark embedding.

The watermark generation process takes sensory data after encryption $E_i$ as input from the sensor node $S_i$ and generates a watermark $W_i$. The time and sensor *ID* are also used as inputs to a keyed-hash message authentication code (*HMAC*) to produce the final watermark. The final is calculated as:

$$W_i = Hash((E_i \mid\mid Time \mid\mid S_iID), k_{cj}) \tag{6}$$

After that, the watermark is embedded at random redundant spaces of the encrypted data $E_i$ as shown in Figure 5. These redundant spaces are determined by using a pseudo-random number generator.

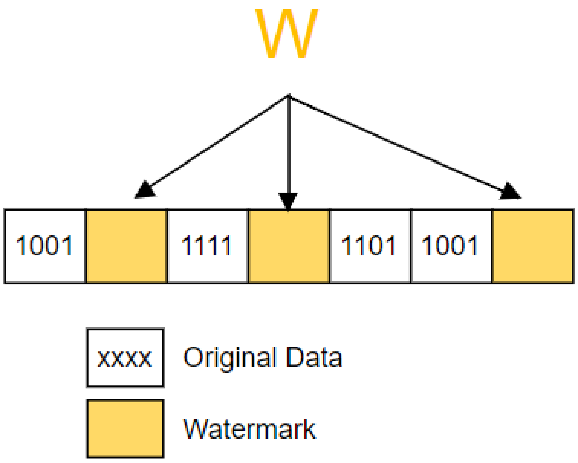


Figure 5: Watermark embedding

There are many ways to build pseudo-random generators suitable for use with our system. But for simplicity, we take the linear congruential generator (LCG) [30] in the following equation to compute each position of $W_i$

$$X_{n+1} = (a_j X_n + c_j) mod\ m_j \tag{7}$$

Here, $m_j > 0$ is a modulus, $a_j$ $(0 < a_j < m_j)$ is the multiplier, $c_j$ $(0 \leq c_j < m_j)$ is the increment, and $X_n$ $(0 \leq X_n < m_j)$ is the current seed. Each generated $X_{n+1}$ is used as a pseudo-random number and becomes the new seed. Each node computes three values of $X_{n+1}$ to use them as the positions of $W_i$.

To avoid every cluster of having the same number generated, each cluster $j$ has different parameters for its pseudo-random number generator.

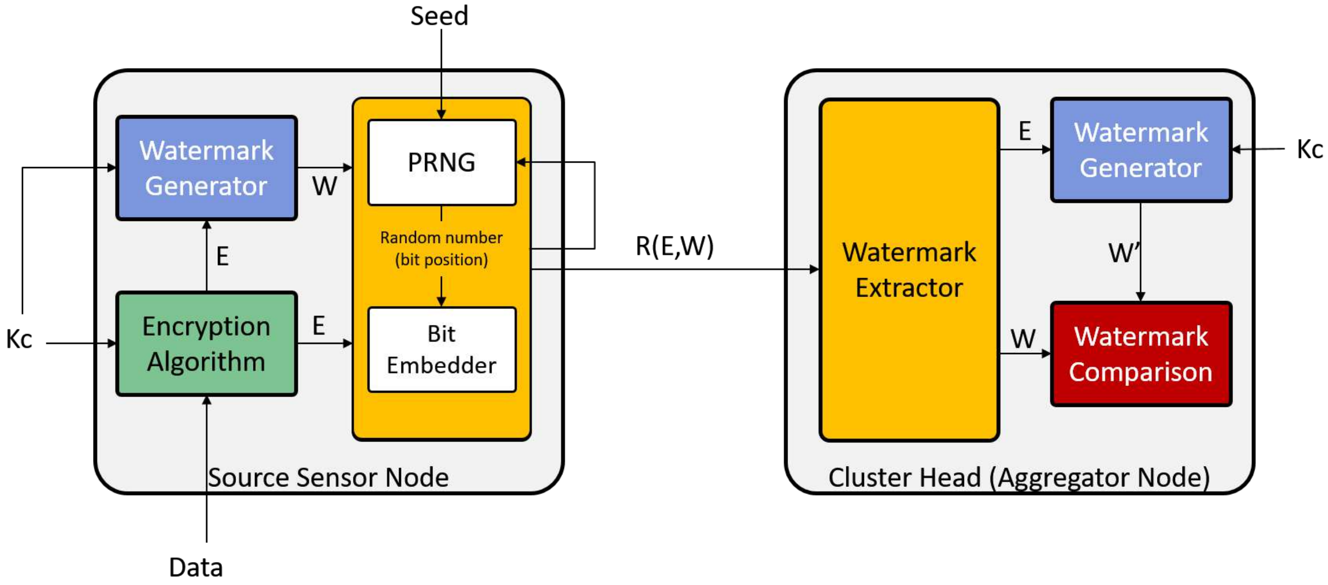


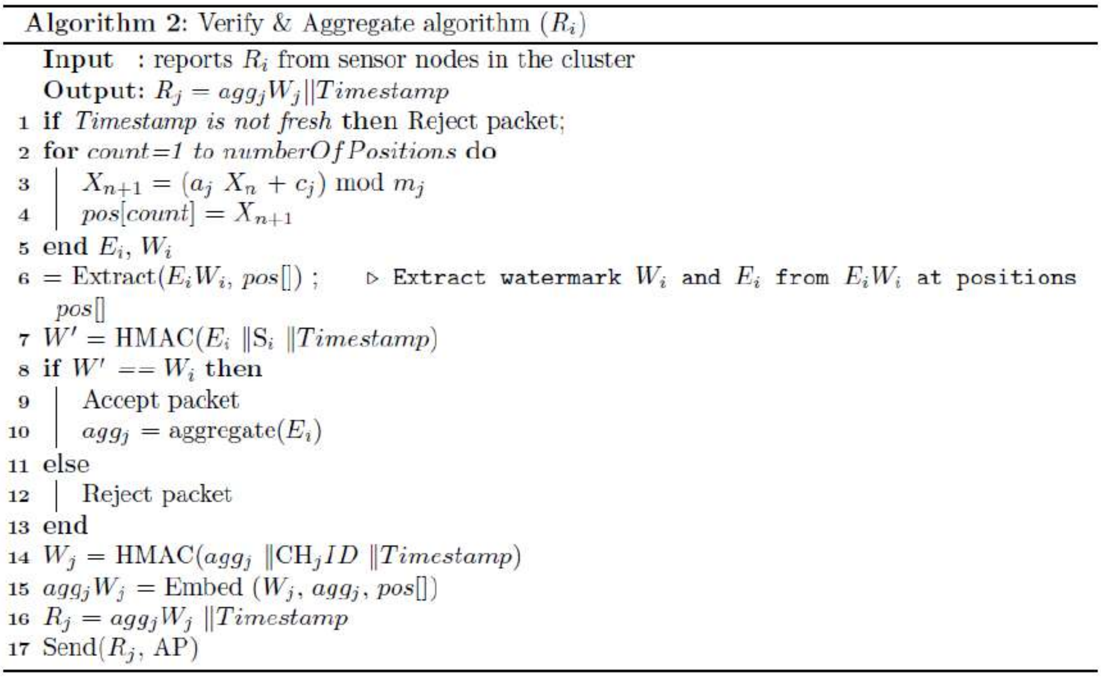
Figure 6: Block diagram of RWFS showing the modules of the sensor node and cluster head. It also shows the flow of data between the modules

*Phase 3: Verifying and Aggregation*

After receiving reports from sensing nodes in cluster *j*, *CH*j starts checking and verifying the integrity of each received report *R*i. It needs to check the embedded watermark and tries to regenerate it by using the shared secret key $K_{cj}$. Therefore, the verification phase relies on three conditions:

1. The freshness of the timestamps $Time$ attached in each report.
2. Verifying the watermark embedded within the measurement.
3. Having $T$ or more reports where T is the number of sensor nodes belonging to cluster j.

Cluster heads can know whether or not the report is the newest measurement based on the field $Time$. This helps in detecting replayed false reports. For condition 2 and 3, the cluster head first extracts the watermark by generating the same number of random numbers to know exactly where the watermark was embedded. Next, it extracts $W_i$ from the received $R_i$ to have $E_i$. A new watermark $W'$is generated from the value of $E_i$ received. If $W_i$ and $W'$ are the same, the report is accepted. After that, $CH_j$ have to wait for $T$ reports. After receiving $T$ reports, $CH_j$ aggregates the measurements received to have $agg_j$, generates $W_j$, embed it and then send it to the AP as $\boldsymbol{R_j} = (\boldsymbol{agg_j} \,|\, \boldsymbol{W_j} \,|\, \boldsymbol{Time})$.



```
Algorithm 2: Verify & Aggregate algorithm ($R_i$)
Input   : reports $R_i$ from sensor nodes in the cluster
Output: $R_j = agg_j W_j \| Timestamp$
if $Timestamp$ is not fresh then Reject packet;
for $count$=1 to $numberOfPositions$ do
    $X_{n+1} = (a_j\ X_n + c_j) \bmod m_j$
    $pos[count] = X_{n+1}$
end $E_i$, $W_i$
= Extract($E_i W_i$, $pos[]$) ;     ▷ Extract watermark $W_i$ and $E_i$ from $E_i W_i$ at positions
  $pos[]$
$W'$ = HMAC($E_i$ ‖S$_i$ ‖$Timestamp$)
if $W'$ == $W_i$ then
    Accept packet
    $agg_j$ = aggregate($E_i$)
else
    Reject packet
end
$W_j$ = HMAC($agg_j$ ‖CH$_j$$ID$ ‖$Timestamp$)
$agg_j W_j$ = Embed ($W_j$, $agg_j$, $pos[]$)
$R_j = agg_j W_j$ ‖$Timestamp$
Send($R_j$, AP)
```

Figure 7: Pseudo code of Verifying and Aggregation phase

## 5. Security Analysis

In this chapter, we analyze the confidentiality and integrity properties of our proposed scheme as for the adversary model presented in Section 0

**Theorem 1:** Our scheme is robust against an adversary of **Type 1** and his attack as it provides end-to-end confidentiality of data.

**Proof:** In the proposed scheme, all data transmitted from sensor nodes to the access point are encrypted with a homographic encryption algorithm by using a shared secret key. Therefore, only parties that know the shared secret can decrypt it. In our scheme, only the access point stores that shared secret and can obtain the plain data. So even though an adversary of **Type 1** eavesdrops the transmission between sensor nodes and CH or the transmission between CH and access point**,** he cannot reveal the content of the messages. This way, the scheme provides end-to-end confidentiality.

**Theorem 2:** Our scheme provides end-to-end integrity in the presence of an adversary of **Type 2**.

**Proof:** The scheme introduces a randomized watermarking technique that checks and verifies the integrity of all packets sent through the network. Any malicious act done to the packet will result in the rejection of that packet. In the following, we state how the scheme secures the network from an attack of the adversary of **Type 2**.

**a. Injection Attack:** an adversary needs to generate a valid watermark to embed it into the injected data. This requires him to know the shared secret cluster key to generate it. And even if the shared secret is somehow disclosed, he will then need to know the random places to embed it. This requires the knowledge of the secret PRNG parameters. This way, all injected false data are rejected either at the CH or the access point.

**b. Replay Attack:** an adversary resends old messages that he captures from the past. This will result in message drops as all packets contain a timestamp to ensure the freshness of the packet.

**c. Modification Attack:** an attacker can modify the data sent and still looks valid. That is due to the homomorphic encryption function properties. For example, a message $m$ can be altered to $m + x$ where x is the modified value. Recipient can decrypt the message and get the modified value. In our scheme, the watermark is computed from the encrypted message and embedded inside the payload. So, any alteration done to the encrypted data will result in a fail of verification of the watermark.

**Theorem 3:** Our scheme is robust against an internal adversary that can compromise a node even if he can control a node or CH from the network.

**Proof:** In the proposed scheme, each cluster has its own security parameter. Therefore, even when an adversary compromises a sensor node or a CH, he will be able to manipulate and control one cluster alone. The access point verifies all clusters data individually and accepts the valid ones and rejects the other.

**Theorem 4:** Our scheme provides en-route filtering for data injected into the network.

**Proof:** The scheme verifies every packet from sensor nodes at CH. Therefore, any injected data at any level will be filtered at the next top level. The end-to-end integrity that is provided by other schemes is not sufficient and can waste energy when forwarding false packets to the access point without verifying its integrity.

## 6. Simulation and Results

In this chapter, we evaluate the performance of the RWFS scheme against the work presented in [12]. We compare their performance regarding filtering capacity, average energy consumption, network lifetime, network overhead and delay.

The simulations of both schemes were done using MATLAB. A heterogeneous WSN containing of $n = 100$ nodes is considered in this simulation. The value of $m = 0.1$ which means there are 10% advanced nodes containing $a$ times more energy than the normal nodes. The nodes were randomly deployed in a $100m \times 100m$ topological area and one access point located at the center of the area $(50m, 50m)$. To unify the simulation of the two schemes, we used the same clustering algorithm (E-DEEC). Table  shows all the parameters used in this simulation.

Table 2: Simulation Parameters

| Parameter | Value |
|---|---|
| Topographical Area | 100 m × 100 m |
| Access point location | (50, 50) |
| Number of nodes | 100 |
| Fraction of advanced nodes $(m)$ | $m = 0.1$ |
| Initial energy of normal node $(E_0)$ | $E_0$= 0.5 J |
| $E_{elec}$ | 50 nJ/bit |

| Parameter | Value |
|---|---|
| $E_{fs}$ | 10 pJ/bit/m$^2$ |
| $E_{mp}$ | 0.0013 pJ/bit/m$^2$ |
| $E_{DA}$ | 5 nJ/bit |
| Packet header size | 150 bits |

### *6.1. Filtering Efficiency*

Filtering efficiency evaluates the ratio of the number of detected malicious packets to the total number of malicious packets. In Figure 8, we have compared the filtering efficiency between the new proposed scheme (RWFS) and Cui, et al. work done in [12].

We have assumed that the attacker knows the MAC method used to generate the watermark in RWFS and the MAC method used to generate the MAC in Cui, et al. scheme. Therefore, the attacker can only randomize values that are secret parameters of each scheme.

For this factor, we have injected the network with different numbers of malicious packets and wanted to know how many packets detected and rejected by the scheme. From the results in Figure 8, we can see that our scheme is better with an accuracy of 99.95% rejection in case of injecting 5000 malicious packets against 99.77% for Cui et al. scheme. This percentage of filtering efficiency proves that the scheme can perform well against any number of injected packets.

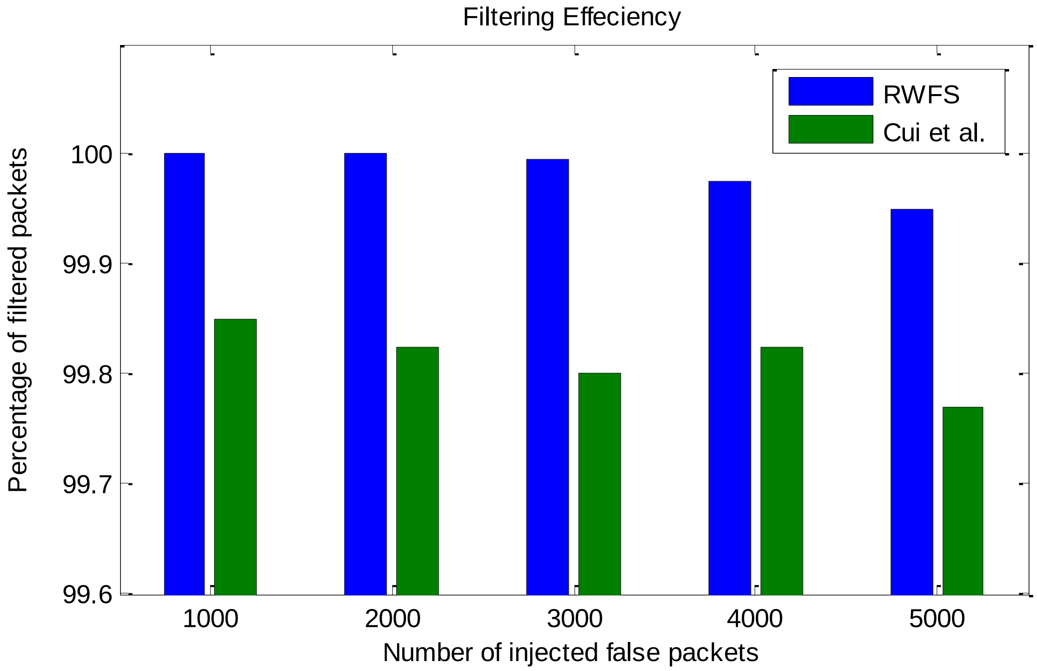


Figure 8: Filtering efficiency analysis between RWFS (proposed scheme) and Cui et al. scheme. The bar graphs show the percentage of filtered malicious packets is shown for a different scenario where a different number of false packets were injected.

### *6.2. Network overhead*

To achieve a fair comparison between our scheme and Cui, et al. scheme [12], we have unified the sizes of common parameters. We have assumed that packets sent during the "Setup and Key Management" phase in RWFS and the ones sent during "Setup" phase in Cui, et al. scheme are both of equal sizes (4000 bits). In addition, both sensor nodes' identities and timestamps were set to 8 and 64 bits respectively.

In RWFS, a packet contains a cipher-text $c_i$ of size 16 bits, an embedded watermark of size 12 bits, the node's identifier and a timestamp, which makes a total of 100 bits of payload per packet. While in Cui, et al. scheme, the packet contains a cipher-text $c_i$ of size 16 bits, a 48 bits homomorphic MAC, a MAC of size 160 bits, a node's identifier and a timestamp. This makes the size of a packet to be 304 bits.

The increase of the packet size is clearly reflected in the overall network overhead. From the comparison results in Figure 9, we can find that our scheme provides a great reduction of network overhead compared to the results of Cui, et al. work during a duration of 10k rounds. Cui, et al. scheme transmit a total of 60% more bytes than our scheme. The reduction of transmitted packets can be directly linked to a longer network lifetime.

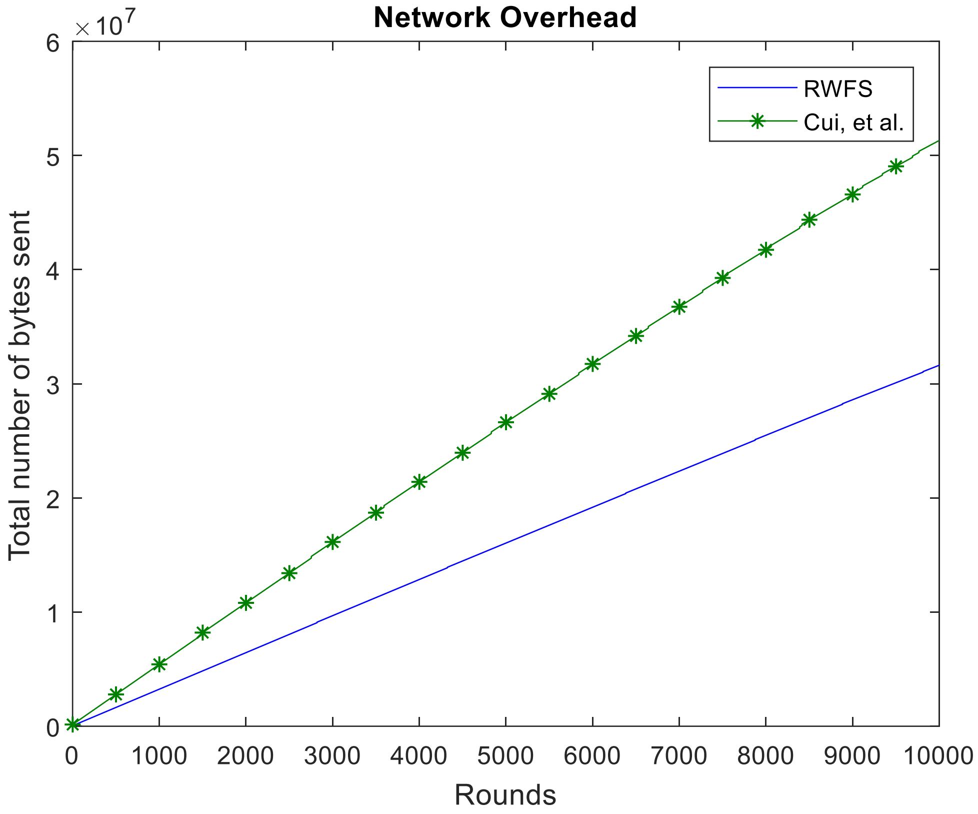


Figure 9: Network overhead analysis between RWFS (proposed scheme) and Cui, et al. scheme where the two graphs show the total number of packets sent during 10k rounds.

### 6.3. Average Energy Consumption

Energy consumption is one of the main factors that matter when developing a new protocol or scheme for WSNs. To evaluate this factor, we have computed the average remaining energy of the whole network after each round. The results are shown in Figure 10.

From Figure 10, we can see that our scheme performs better in terms of saving the energy of the network. RWFS can save as up to 85 % more energy than Cui, et al. scheme. This is a direct impact of reducing the packet size. As the larger the packet, the more energy consumed by a node to send and receive packets.

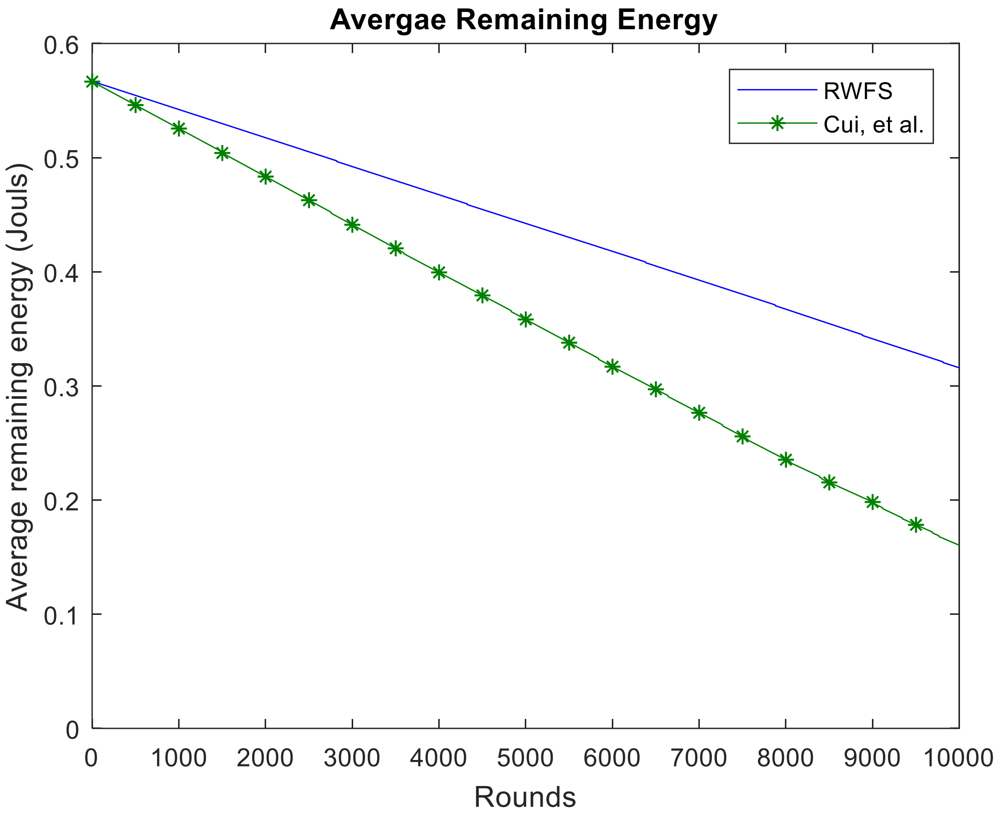


Figure 10: Average remaining energy analysis between RWFS (proposed scheme) and Cui, et al. scheme. The average energy of all nodes in the network is calculated at each round in a duration of 10k rounds.

### *6.4. Network Lifetime*

Network lifetime can be defined as "*the time at which the first network node runs out of energy to send a packet*" [20]. Therefore, we have investigated the number of dead nodes after a number of rounds to get a clear indication of the network lifetime of both schemes. Figure 11 shows that after 1500 rounds, a node was out of energy in Cui, et al. scheme. While RWFS reached 2800 rounds before the first node was dead. We can also see that after 10K rounds, only seven nodes were dead in our scheme, which is a third of the number of dead nodes that were in Cui, et al. scheme.

This implies that our scheme is more energy-efficient than the one presented by Cui, et al. This can be due to the fact that our scheme consumes less energy per round and uses a smaller packet.

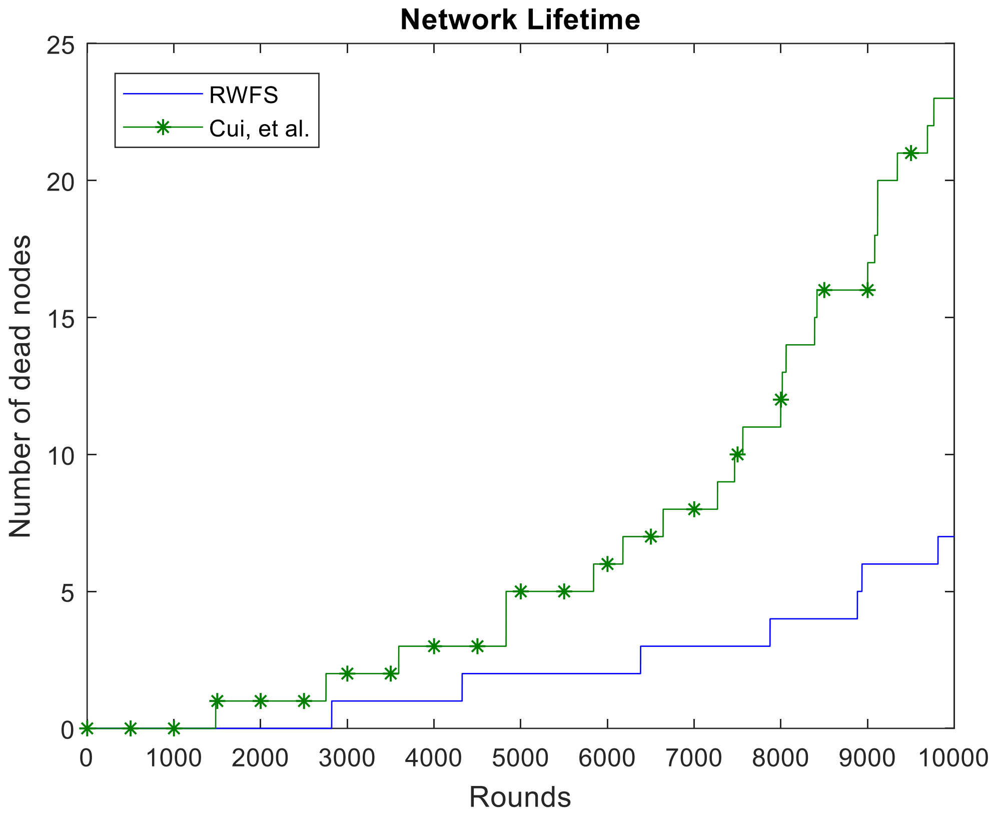


Figure 11: Network lifetime analysis between RWFS (proposed scheme) and Cui, et al. scheme where the number of dead nodes is calculated at each round in a duration of 10k rounds.

### *6.5. Delay*

We define delay here as the required time to complete a whole round. It is the time to sense and process data at leaf nodes, verify and aggregate data at the cluster heads and finally verify and decrypt data at the access point.

From the results in Figure 12, we can see that the average delay of one round of RWFS is 0.086 seconds which is 65% lesser than the delay of Cui, et al. scheme. This can be explained since (Cui, et al.) work have an additional homomorphic MAC (H-MAC) that is computed for each sensed data, along with a MAC. One calculation of H-MAC alone is found to be 1.3 *ms* on average. This makes RWFS $1.3 \times n$ *ms* faster than Cui, et al. scheme.

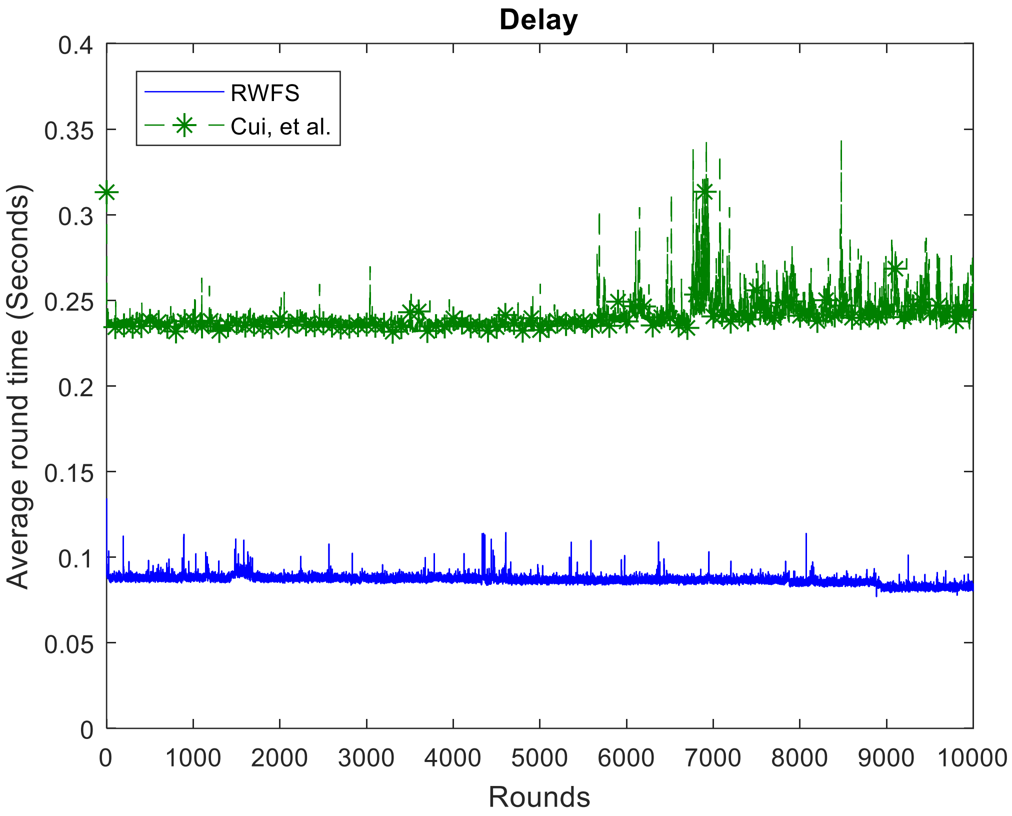


Figure 12: Delay analysis between RWFS (proposed scheme) and Cui, et al. scheme. The Average execution time for each round was measured in a total of 10k rounds.

## 7. Conclusion

IoT applications are playing a major role in building new solutions for various real-world problems. They connect physical sensors, field control centers, and cloud systems to enable reliable and smart applications. Recently, IoT applications are experiencing a tremendous industrial interest due to their advantages. However, physical attacks, such as false injection attacks, are considered one of the main threats for the security of the devices in the Internet of Things. False data injection attacks are considered very crucial in sensitive networks where any small alteration in the data sensed measurement could lead to a sever action. Therefore, it is very important to employ a scheme that provides filtering ability against false data injection attacks taking into consideration all the limitation of the sensors that make the networked system. Several attempts have been developed to design such a scheme, but all of them face some challenges. The common limitation of the existing solutions is energy-related issues.

This paper presented a lightweight, randomized watermarking filtering scheme (RWFS) that provides data integrity, along with data confidentiality, in wireless sensor network for IoT applications. The primary goal of this proposed scheme is to be an energy efficient mechanism for securing such networks against data injection attacks. We employ a homomorphic encryption algorithm to protect end-to-end data confidentiality and a watermark to achieve en-route data filtering. The scheme uses a watermark that is generated and embedded with the original data sent by all source nodes in the network. The watermark is generated based on four features including encrypted payload, the secret shared key of the cluster, sensor's identifier and data capturing time of the individual sensor node. This watermark is then embedded into random places within the packet. These random places are generated using a PRNG function. After this, the watermarks are verified by the respective cluster head to ensure the integrity of the packets. For testing purpose, we have simulated our scheme in comparison with Cui et al. scheme. Cui et al. work uses a homomorphic

MAC function to provide the end-to-end integrity which is computationally expensive and takes more time during generation. Whereas our experimental results proved that RWFS scheme achieves better computational efficiency and consumes less energy.

**Author Contributions:** Conceptualization, Arwa Alromih, Mznah Al-Rodhaan and Yuan Tian; Formal analysis, Arwa Alromih; Funding acquisition, Arwa Alromih; Investigation, Arwa Alromih; Methodology, Arwa Alromih, Mznah Al-Rodhaan and Yuan Tian; Software, Arwa Alromih; Supervision, Mznah Al-Rodhaan and Yuan Tian; Visualization, Arwa Alromih; Writing – original draft, Arwa Alromih; Writing – review & editing, Arwa Alromih, Mznah Al-Rodhaan and Yuan Tian.

**Acknowledgments:** This research was supported by a grant from the "Research Center of the Female Scientific and Medical Colleges", Deanship of Scientific Research, King Saud University.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Gubbi, J.; Buyya, R.; Marusic, S.; Palaniswami, M., Internet of Things (Iot): A Vision, Architectural Elements, and Future Directions. *Future generation computer systems* **2013,** *29* (7), 1645-1660.
2. Lazarescu, M. T., Wireless Sensor Networks for the Internet of Things: Barriers and Synergies. In *Components and Services for Iot Platforms*, Springer: 2017; pp 155-186.
3. Mahgoub, I.; Ilyas, M., *Sensor Network Protocols*. CRC press: 2016.
4. Kui, R.; Wenjing, L.; Yanchao, Z., Leds: Providing Location-Aware End-to-End Data Security in Wireless Sensor Networks. *IEEE Transactions on Mobile Computing* **2008,** *7* (5), 585-598.
5. Illiano, V. P.; Lupu, E. C., Detecting Malicious Data Injections in Wireless Sensor Networks: A Survey. *ACM Comput. Surv.* **2015,** *48* (2), 1-33.
6. Granjal, J.; Monteiro, E.; Silva, J. S., Security in the Integration of Low-Power Wireless Sensor Networks with the Internet: A Survey. *Ad Hoc Networks* **2015,** *24*, 264-287.
7. Hameed, K.; Khan, M. S.; Ahmed, I.; Ahmad, Z. U.; Khan, A.; Haider, A.; Javaid, N. In *A Zero Watermarking Scheme for Data Integrity in Wireless Sensor Networks*, 19th International Conference on Network-Based Information Systems (NBiS), 2016 Ostrava, Czech Republic, IEEE: Ostrava, Czech Republic, 2016; pp 119-126.
8. Bartariya, S.; Rastogi, A., Security in Wireless Sensor Networks: Attacks and Solutions. *environment* **2016,** *5* (3).
9. Yu, Z.; Guan, Y. In *A Dynamic En-Route Scheme for Filtering False Data Injection in Wireless Sensor Networks*, SenSys, 2005; pp 294-295.
10. Di Pietro, R.; Michiardi, P.; Molva, R., Confidentiality and Integrity for Data Aggregation in Wsn Using Peer Monitoring. *Security and Communication Networks* **2009,** *2* (2), 181-194.
11. Kumar, M.; Verma, S.; Lata, K., Secure Data Aggregation in Wireless Sensor Networks Using Homomorphic Encryption. *International Journal of Electronics* **2015,** *102* (4), 690-702.
12. Cui, J.; Shao, L.; Zhong, H.; Xu, Y.; Liu, L., Data Aggregation with End-to-End Confidentiality and Integrity for Large-Scale Wireless Sensor Networks. *Peer-to-Peer Networking and Applications* **2018,** *11* (5), 1022-1037.
13. Xinyu, Y.; Jie, L.; Wei, Y.; Moulema, P. M.; Xinwen, F.; Wei, Z., A Novel En-Route Filtering Scheme against False Data Injection Attacks in Cyber-Physical Networked Systems. *Computers, IEEE Transactions* **2015,** *64* (1), 4-18.
14. Fang, J.; Potkonjak, M. In *Real-Time Watermarking Techniques for Sensor Networks*, Electronic Imaging 2003, International Society for Optics and Photonics: 2003; pp 391-402.
15. Tiwari, A.; Chakraborty, S.; Mishra, M. K. In *Secure Data Aggregation Using Irreversible Watermarking in Wsns*, Confluence 2013: The Next Generation Information Technology Summit (4th International Conference), Noida, India, IET: Noida, India, 2013; pp 330-336.
16. Boubiche, D. E.; Boubiche, S.; Bilami, A., A Cross-Layer Watermarking-Based Mechanism for Data Aggregation Integrity in Heterogeneous Wsns. *IEEE Communications Letters* **2015,** *19* (5), 823-826.
17. Sun, X.; Su, J.; Wang, B.; Liu, Q., Digital Watermarking Method for Data Integrity Protection in Wireless Sensor Networks. *International Journal of Security and Its Applications* **2013,** *7* (4), 407-416.
18. Lalem, F.; Muath, A.; Bounceur, A.; Euler, R.; Laouamer, L.; Nana, L.; Pascu, A. C. In *Data Authenticity and Integrity in Wireless Sensor Networks Based on a Watermarking Approach*, The 29th International Florida Artificial Intelligence Research Society, Key Largo, Florida, USA., Key Largo, Florida, USA., 2016.

19. Ren, Y.; Cheng, Y.; Wang, J.; Fang, L. In *Data Protection Based on Multifunction Digital Watermark in Wireless Sensor Network*, International Carnahan Conference on Security Technology (ICCST), 2015, Taipei, Taiwan, IEEE: Taipei, Taiwan, 2015; pp 37-41.
20. Guan, T.; Chen, Y. In *A Node Clone Attack Detection Scheme Based on Digital Watermark in Wsns*, IEEE International Conference on Computer Communication and the Internet (ICCCI), 2016 Wuhan, China, IEEE: Wuhan, China, 2016; pp 257-260.
21. Shi, X.; Xiao, D., A Reversible Watermarking Authentication Scheme for Wireless Sensor Networks. *Information Sciences* **2013,** *240*, 173-183.
22. Ding, Q.; Wang, B.; Sun, X.; Wang, J.; Shen, J., A Reversible Watermarking Scheme Based on Difference Expansion for Wireless Sensor Networks. *International Journal of Grid Distribution Computing* **2015,** *8* (2), 143-154.
23. Li, X.; Zhong, Y.; Liao, F.; Li, R., An Improved Watermarking Scheme for Secure Data Aggregation in Wsns. *Applied Mechanics & Materials* **2014,** *556-562*, 6298-6301.
24. Saini, P.; Sharma, A. K. In *E-Deec- Enhanced Distributed Energy Efficient Clustering Scheme for Heterogeneous Wsn*, 2010 First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010), Solan, India, 28-30 Oct. 2010; Solan, India, 2010; pp 205-210.
25. Heinzelman, W. R.; Chandrakasan, A.; Balakrishnan, H. In *Energy-Efficient Communication Protocol for Wireless Microsensor Networks*, Proceedings of the 33rd annual Hawaii international conference on System sciences., IEEE: 2000; p 10 pp. vol. 2.
26. Pourpeighambar, S. B.; Aminian, M.; Sabaei, M. In *Energy Efficient Data Aggregation of Moving Object in Wireless Sensor Networks*, Telecommunication Networks and Applications Conference (ATNAC), 2011 Australasian, IEEE: 2011; pp 1-8.
27. Bushnag, A.; Abuzneid, A.; Mahmood, A., Source Anonymity in Wsns against Global Adversary Utilizing Low Transmission Rates with Delay Constraints. *Sensors* **2016,** *16* (7), 957.
28. Dai, H.-N.; Wang, Q.; Li, D.; Wong, R. C.-W., On Eavesdropping Attacks in Wireless Sensor Networks with Directional Antennas. *International Journal of Distributed Sensor Networks* **2013,** *9* (8), 760834.
29. Castelluccia, C.; Chan, A. C.-F.; Mykletun, E.; Tsudik, G., Efficient and Provably Secure Aggregation of Encrypted Data in Wireless Sensor Networks. *ACM Trans. Sen. Netw.* **2009,** *5* (3), 1-36.
30. Knuth, D. E., *The Art of Computer Programming*. Addison-Wesley: 1997; Vol. 2: Seminumerical Algorithms.