applied sciences

MDPI

Article

An Anonymous Mutual Authenticated Key Agreement Scheme for Wearable Sensors in Wireless Body Area Networks

Chien-Ming Chen [1], Bing Xiang [1], Tsu-Yang Wu [2,3] and King-Hang Wang [4,*]

[1] Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China; chienming.taiwan@gmail.com (C.-M.C.); xiangbin.hit@qq.com (B.X.)
[2] Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, China; wutsuyang@gmail.com
[3] National Demonstration Center for Experimental Electronic Information and Electrical Technology Education, Fujian University of Technology, Fuzhou 350118, China
[4] Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong, China
* Correspondence: kevinw@cse.ust.hk; Tel.: +852-2358-8839

Received: 31 May 2018; Accepted: 21 June 2018; Published: 2 July 2018

Abstract: The advancement of Wireless Body Area Networks (WBAN) have led to significant progress in medical and health care systems. However, such networks still suffer from major security and privacy threats, especially for the data collected in medical or health care applications. Lack of security and existence of anonymous communication in WBAN brings about the operation failure of these networks. Recently, Li et al. proposed a lightweight protocol for wearable sensors in wireless body area networks. In their paper, the authors claimed that the protocol may provide anonymous mutual authentication and resist against various types of attacks. This study shows that such a protocol is still vulnerable to three types of attacks, i.e., the offline identity guessing attack, the sensor node impersonation attack and the hub node spoofing attack. We then present a secure scheme that addresses these problems, and retains similar efficiency in wireless sensors nodes and mobile phones.

Keywords: security; anonymity; WBAN; wearable sensors; cryptanalysis

1. Introduction

The advancement of electromedical technology has led to new research topics associated with wireless body area networks (WBANs). A wireless body area network (WBAN) is formed by a medication information system and various wearable sensors attached to the patient's body. Integration of WBAN with modern cloud and sensor technologies offers huge improvement in the efficiency and functionality of medical and health care systems. For instance, after the ischemic stroke, patients would require a long-term electrocardiographic monitoring [1]. They suffer from the sleep apnea, and, consequently, require to wear a portable monitor while sleeping [2]. A WBAN-enabled environment allows patients to enjoy the same quality of life without being tangled by the sensor wires. To provide a comprehensive and real-time health assessment to the patient, sensed data may be transmitted to the clouds.

A WBAN architecture is generally constituted of three layers, as shown in Figure 1. This architecture is composed of three types of nodes, first level nodes, second level nodes and a hub node. The first level node, e.g., a smartphone, acts as an intermediate node and forwards the data to the hob node. The second level nodes normally refer to the nodes or wearable devices situated in the body of human, sending the sensing information to a first level node. The hub node a local server or a remote cloud that analyses and manages the sensed data.

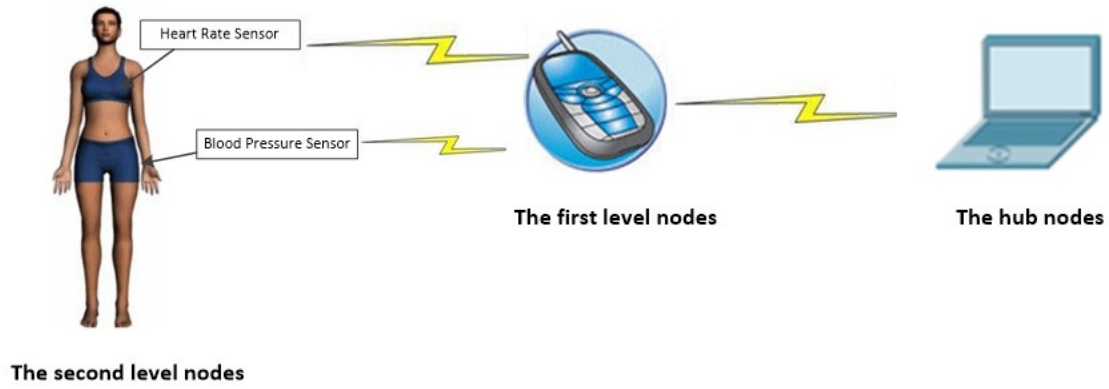

Figure 1. Architecture of a medical WBAN.

Despite the WBANs being endowed with the simplicity and high efficiency, they suffer from low security so that the transmitted data contain the health information of the user which is typically highly sensitive. The need of finding a secure solution for the network is immediate as the security association in the 802.15.6 standard is in doubt [3]. To guarantee a secure WBAN, a secure authentication key agreement protocol should be executed in advance of the communication. We argue that this protocol still requires the user anonymity. Consider a user wearing a portable electrocardiographic monitor to keep track of his cardio health, where the cardio data are appropriately encrypted. The privacy of a known data transfer channel is compromised so that the electrocardiographic monition has been related to a cardio problem through other users.

According to the previously reported works, e.g., [3,4], the authentication key agreement protocol of the WBAN shall provide the data secrecy, user anonymity, session unlinkability, mutual authentication, forward secrecy, resilient to online/offline dictionary attack, resilient to replay attack, and resilient to man-in-the-middle attack. Due to a few reasons, we should not use generic authentication key agreement protocols [5] or lightweight protocols for the general purpose short distance communications [6] in WBANs. Firstly, the specific architecture of the WBAN includes three tiers with multiple first level nodes whose most generic protocols are not optimized in this setting. Some first level nodes may be restricted in terms of power or computation ability so that a heavy computation is not possible. Furthermore, some generic authentication protocols may not offer the user anonymity as their protocol design requirement. However, in a WBAN, the identity of the patient should be concealed while being diagnosed with a WBAN.

WBANs share some similar properties with Hierarchical Wireless Sensor Networks (HWSN). The valuable experience established in the HWSN research area has in turn led to the fast development of WBANs. Wang et al. [7] has summarized some early advancement in the authentication protocol of HWSNs. However, conventional HWSNs assume a large-scale network and are more concerned about the battery power than the security and user's privacy. As of today, there has been no direct applicable of HWSN to WBAN.

Recently, various authentication and key agreement protocols for WBANs have been proposed. In 2009, Keoh et al. [8] has reported a protocol using an on synchronized LED blinking pattern and keychains that provides a visual confirmation of the sensor pairing. Later, Liu et al. [9] presented another protocol using both public key and secret key cryptography in the authentication. In 2014 Liu et al. [10] improved the anonymity over their previous work and presented a protocol focusing on the communication between the first level and second level nodes using the elliptic curve cryptogrpahy and bilinear map. Moreover, the anonymity of the scheme was broken by Zhao in 2014 [11]. Zhao and, subsequently, Wu et al. [12] presented their protocols to overcome some weakness founded in previous works. Those protocols however require the use of public key cryptography (either elliptic curve cryptography or bilinear pairing) in the sensor node yielding a heavy computation and storage

bundle [13]. In order to save resources and ensure anonymity, Shen et al. [14] proposed a cloud-aided lightweight authentication protocol. Their protocol ensures that the network manager cannot realize the user's real identity in the authentication phase.

The sensors attached on the human bodies have direct access to the physiological signals of the person. As a result, following the electrocardiogram (ECG) or photoplethysmogram (PPG), the use of these physiological signals may be used to generate keys of the communication [15–17]. Such an approach is quite novel and can be possibly developed in good applications after its robustness and security may be verified in a larger scale or experiments. Unlike secrets, and like passwords or pre-loaded secret keys, the physiological signal may not be necessarily kept away from the attackers.

In 2017, Li and his colleagues proposed a lightweight mutual authentication and key agreement protocol with anonymity for the WBAN [4]. They claimed that their protocol provides anonymity and may be secure against various types of attacks. However, this study demonstrates that Li's protocol is not secure while the first level node is being compromised. In addition, their approach fails to provide the node anonymity so that an attacker is able to track a second level node. To overcome these shortcomings, we provide a simple but effective amendment for the protocol. The repaired protocol is secured against impersonation attacks, replay attacks, and man-in-the-middle attacks. It also provides better anonymity of the WBAN users.

The organization of the paper is as follows. Section 2 reviews the Li's scheme. In Section 3, we show the insecurity of their scheme. Next, an improvement scheme will be presented in Section 4. We then provide some security analysis on the improved scheme, and finally conclude the paper.

2. Review of the Li's Protocol

In this section, we briefly review the Li's protocol [4]. Figure 2 shows the architecture of this protocol, which consists of three level nodes, i.e., a hub node (*HN*), a first level nodes (*FN*) and some second level nodes (*SN*). The second level nodes are some wearable sensors to be attached to the human body. Usually, these *SN* are resource-constrained with limited computational and communicational power. They report sensed data to a first level node (*FN*) via a public channel. A *FN* is an intermediate node between *SN* and *FN*. It may be considered as a smart phone or a smart watch, providing good communication and computation ability and coordinating a set of *SN* attached to the same human body. Next, the *FN* forwards the received sensed data to a hub node (*HN*), which was formed by rich resources and may be installed on a database.

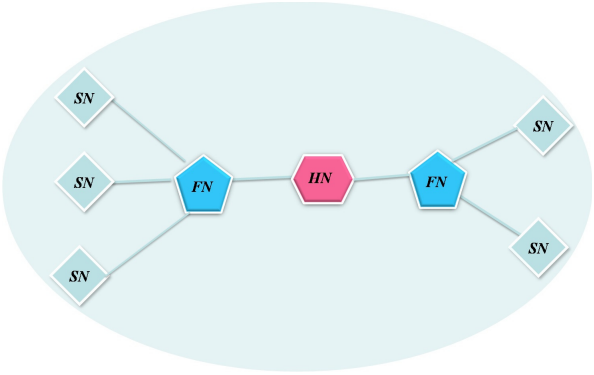

Figure 2. Architecture of Li's protocol [4].

Such a protocol is composed of two phases as follows, the registration phase and the authentication phase. In the registration phase, a system administrator registers and initializes the *HN*, *FN*, and *SN*. In the authentication phase, an *SN* attempts to setup a secure connection in the network while authenticate the identity of the *HN* and being authenticated by the *HN*.

2.1. Registration Phase

In this phase, an *HN* generates a unique secret key, k_{HN}, and securely stores it in its memory. In addition, each second level node is registered individually.

Once a second node N is being registered, the following steps are performed:

1. A unique secret identity id_N is generated for the N which is also used as the secret key of the N.
2. A unique identity id'_N is generated for the *FN*. (It is not explicit in their article that would another id'_N be generated or not when another *SN* is registered. However, if different id'_N is generated for the *SN* that will immediately fail the *SN*'s traceability since the unencrypted id'_N is sent over the air every time the *SN* attempts to connect to the server).
3. A secret parameter k_N is generated for the N.
4. The system computes $a_N = id_N \oplus h(k_{HN}, k_N)$ and $b_N = k_{HN} \oplus a_N \oplus k_N$.
5. The *FN* stores the tuple $\langle id'_N, id_N, a_N, b_N \rangle$ in its memory.
6. The N stores the tuple $\langle id_N, a_N, b_N \rangle$ in its memory.
7. The *HN* stores the (id'_N) in its memory.

Note that k_N is not required to be stored in the sensor node *SN* or at the hub node *HN*.

2.2. Authentication Phase

In this phase, the N establishes a session key with the *HN* through the *FN* as follows. The whole process is given in Figure 3.

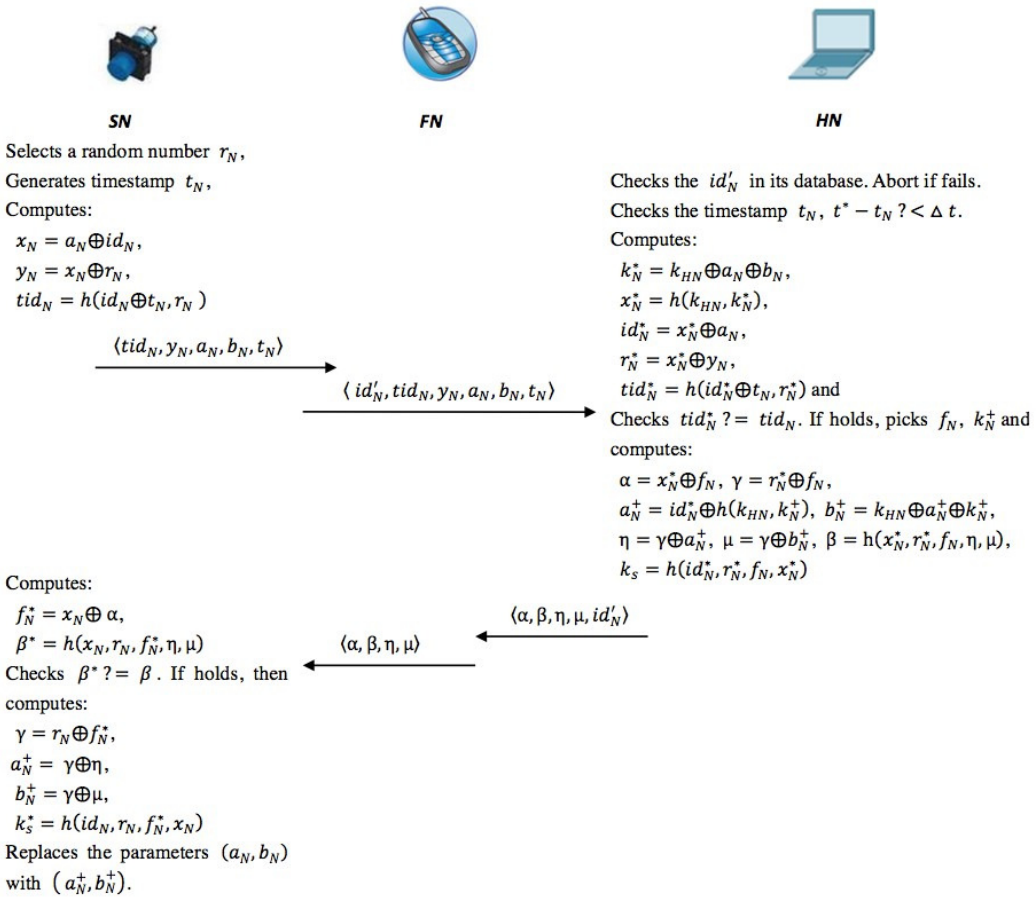

Figure 3. Li's protocol.

1. A second level node N selects a random number r_N and computes

$$x_N = a_N \oplus id_N, \tag{1}$$
$$y_N = x_N \oplus r_N, \tag{2}$$
$$tid_N = h(id_N \oplus t_N, r_N), \tag{3}$$

where t_N is the current timestamp. Next, the N sends $\langle tid_N, y_N, a_N, b_N, t_N \rangle$ to the FN.

2. After receiving the message from the N, the FN places his identity, id'_N, in the message and forwards the message $\langle id'_N, tid_N, y_N, a_N, b_N, t_N \rangle$ to the HN.
3. Once receiving messages from the FN, the HN first checks the id'_N in its database. The process will be terminated if fails. Then, the HN checks the timestamp t_N by judging $t^* - t_N \stackrel{?}{<} \delta t$, where t^* is the time when the message is received, with δt being the maximum transmission delay. Next, the HN computes the following:

$$k^*_N = k_{HN} \oplus a_N \oplus b_N, \tag{4}$$
$$x^*_N = h(k_{HN}, k^*_N), \tag{5}$$
$$id^*_N = x^*_N \oplus a_N, r^*_N = x^*_N \oplus y_N, \tag{6}$$
$$tid^*_N = h(id^*_N \oplus t_N, r^*_N), \tag{7}$$

4. which checks whether $tid^*_N \stackrel{?}{=} tid_N$. If the equation holds, the HN ensures that the N is legal. The HN picks temporary secret parameters f_N, k^+_N and continues to compute the following:

$$\alpha = x^*_N \oplus f_N, \tag{8}$$
$$\gamma = r^*_N \oplus f_N, \tag{9}$$
$$a^+_N = id^*_N \oplus h(k_{HN}, k^+_N), \tag{10}$$
$$b^+_N = k_{HN} \oplus a^+_N \oplus k^+_N, \tag{11}$$
$$\eta = \gamma \oplus a^+_N, \tag{12}$$
$$\mu = \gamma \oplus b^+_N, \tag{13}$$
$$\beta = h(x^*_N, r^*_N, f_N, \eta, \mu). \tag{14}$$

5. Finally, the HN stores the session key $k_s = h(id^*_N, r^*_N, f_N, x^*_N)$ and sends the message $\langle \alpha, \beta, \eta, \mu, id'_N \rangle$ to the FN.
6. Once the FN receives the message from the HN, it drops his identity id'_N and sends the message $\langle \alpha, \beta, \eta, \mu \rangle$ to the N.
7. Now, the N computes $f^*_N = x_N \oplus \alpha, \beta^* = h(x_N, r_N, f^*_N, \eta, \mu)$ and checks $\beta^* \stackrel{?}{=} \beta$ to determine whether the HN is legal or not. The authentication process will terminate if the equation does not hold. Then, the N computes $\gamma = r_N \oplus f^*_N, a^+_N = \gamma \oplus \eta$, and $b^+_N = \gamma \oplus \mu$. Afterwards, the N stores the session key $k^*_s = h(id_N, r_N, f^*_N, x_N)$ and replaces the parameters (a_N, b_N) with the parameters (a^+_N, b^+_N).

3. Cryptanalysis of the Li's Protocol

This section shows that the protocol proposed by Li, and his colleagues, is vulnerable to three types of attacks, i.e., offline identity guessing attacks, sensor node impersonation attacks and hub node spoofing attacks.

3.1. The Adversary Model

We assume the adversary is capable of performing the following, once being attacked. The first three capabilities are adopted from the Li's paper while the last one is a reasonable extension of their model:

- The adversary can control the communication channel. It means that it may eavesdrop, modify and replay any messages transmitted on the communication channel. This intends to capture the protocol requirements, e.g., resilient to replay the attack, resilient man-in-middle attack, mutual authentication, resilient to online/offline dictionary attack.
- The adversary can capture any sensor node by some ways and further extract the secret data store in a captured node. This intends to capture the ability of mutual authentication and forward secrecy.
- The hub node, *HN*, is always trustworthy. However, an adversary may intrude the *HN*'s database and read and manipulate all the data in the database except for the *HN*'s master key, k_{HN}. This intends to capture the resilient of the hub-node-stolen-database attack where the *HN*'s database is stolen.
- An adversary may intrude a first level node *FN* and read all data stored in it. Assuming that both the bottom level *SN* and the top level *HN* can be compromised by the adversary, the *FN* may not remain unintruded for all the time, especially an *FN* may be viewed as a smart phone or a smart watch which may be easily stolen.

3.2. Vulnerable against Intruding FN Attacks

In the protocol design, an *FN* is mainly served as a intermediate relay. However, during the registration phase, the secret information, e.g., id_N, a_N and b_N are all stored in the *FN*. It is not explicit how these values shall be used in the *FN* according to their paper. It is observed that the *FN* does not have the capability to authenticate an *SN* and to be authenticated by the *HN* on behalf of an *SN*, if the *FN* is responsible to coordinate the *SN*. Nevertheless, this turns out to become a point of vulnerability of the protocol. For an adversary which is able to intrude an *FN*, all *SN* s coordinated by this *FN* are compromised.

3.3. Vulnerable to the Tracking Attack

Li claimed that the protocol allows anonymous communication so that an adversary cannot link any communication session to another session of the same *SN*. However, this claim is not true, based on the following facts.

Every *SN* is registered to the system through one single *FN*. The identity of the *FN*, id'_N, is sent over the air in Step 2 of the authentication phase. Since id'_N would not be changed in the protocol, adversary can be easily associated with two sessions with the same *FN* s. For an *FN* coordinating only one *SN*, the adversary is allowed to link two sessions of the same *SN* by inspecting only Step 2. If the *FN* coordinates more *SN* s, the user's privacy/anonymity does not enhance as in some applications suggested in Li's paper. Consider the medication, where the sensors of a patient are likely to be connected to a single *FN*, e.g., his smart phone. Revealing the identity of the *FN* (smart phone) is even worse than revealing only the identity of an *SN* (a sensor).

In certain applications, an *FN* may coordinate extremely large amount of *SN* s, where the identity of the *SN* is the only concern and an adversary is still able to link two sessions with the same *SN* s. Assuming that the adversary $\mathcal{A}$ captures only the messages sent from the *SN* to *FN* and *FN* to *SN* at the time T_1 and a later time T_2, as

$$\text{Capture at } T_1 : \begin{cases} \langle tid_1, y_1, a_1, b_1, t_1 \rangle \\ \langle \alpha_1, \beta_1, \eta_1, \mu_1 \rangle \end{cases}, \quad \text{Capture at } T_2 : \begin{cases} \langle tid_2, y_2, a_2, b_2, t_2 \rangle \\ \langle \alpha_2, \beta_2, \eta_2, \mu_2 \rangle \end{cases}.$$

To investigate if the messages captured at T_2 is a subsequent login of the messages captured at T_1, the $\mathcal{A}$ simply computes $a_2 \oplus b_2$. If these two sessions are related, this value corresponds to $(\gamma_1 \oplus \eta_1) \oplus (\gamma_1 \oplus \mu_1) = \eta_1 \oplus \mu_1$, which is indeed $k_{HN} \oplus k_N$. Except for an extreme low probability of coincident $(2^{-\text{length}(k_{HN})})$, comparing $a_2 \oplus b_2 \stackrel{?}{=} \eta_1 \oplus \mu_1$ will allow for determining if these two sessions are related.

4. Repairing the Protocol

One of the biggest problems associated with the protocol is that the *FN* does not perform its function in the authentication while it is possessing the secret information of the coordinating *SN*. A simple straightforward approach is to let the *FN* not store any information about the *SN*. Instead, the *FN* only acts as a relay between the *SN* and the *HN*. The protocol will be remaining secure (but not anonymous) even if the *FN* is being compromised. This however does not resolve the vulnerability of the protocl against the tracking attacks. Moreover, this option removes the ability of an *FN* to control other *SN*s, which may not be suitable in some applications.

The security and system requirements may be investigated as follows. The *SN*s assume low computation/communication power; while *FN*s and *HN*s are less constrained, the *SN*s and *HN*s require being mutually authenticated. The *SN* and *FN* should be mutually authenticated where these two authentications may not be necessarily at the same time. Based on these requirements, we propose a simpler repaired protocol exhibiting better security and anonymity.

4.1. Architecture

In our architecture, we maintain the three-level role. However, the communication between an *SN* and an *FN* (*SN-FN*) is different from the communication session between an *SN* and an *HN* (*SN-HN*). A two-party authentication protocol will be described in this section, and the *same protocol* will be used in the case of *SN-FN* and *SN-HN*. In the case of an *SN-HN* communication, the *FN* will be served as a relay to support the communication. The *SN-HN* communication normally takes place when the sensing data is reported to the *HN*. The *SN-FN* communication normally takes place when *FN* manages the *SN* or gathering data from the *SN*. In the case where *FN-HN* communication is required, we assume that general purpose authentication protocols, e.g., [5,18], will be used since both of them have less constraint computation power.

4.2. Description of the Repaired Protocol

As mentioned above, this protocol is a two-party protocol. The reader may assume a duplication of keys for the *SN-FN* and *SN-HN* communications. We call the *UN* an upstream node that represents either an *FN* or an *HN*. Unless it is specified, all variables have the same length as the output of a hash function length(h).

A *SN* should separately register with an *FN* and an *HN*, and two sets of keys are required. Practically, these two registrations may be simultaneously performed via the *FN*, as long as the process is securely accomplished. Assume that the *SN* is registering with either of them, denoted as a *UN*. The *SN* will then be assigned with the followings:

- id_N, a unique secret identity for the *SN*.
- $a_N = id_N \oplus h(k_{UN}, k_N)$, where k_{UN} is the secret key of the *UN*, k_N is a nonce.
- $b_N = a_N \oplus k_{UN} \oplus k_N$.
- $c_N = h(id_N, k_{UN})$.

In this protocol, the *UN* does not require storing any secret information about the *SN*. If the *UN* wishes to keep track of the identity of the *SN*, it may keep a truncated or hashed id_N. The value of the id_N needs to be unique and a bit of id_N may be used to indicate the association with either of *SN-HN* or *SN-FN*, and several bits from the identity of the *UN*.

When the *SN* wishes to initiate a communication with a *UN*, the *SN* will perform the following operations (In case an *FN* wishes to initiate the protocol, the protocol will be preceded by a *Hello* message from the *FN* to the *SN*.). Please also refer to Figure 4.

1. The *SN* generates a random number r_N and a timestamp t_N and computes:

$$x_N = a_N \oplus id_N, \tag{15}$$
$$y_N = x_N \oplus r_N, \tag{16}$$
$$tid_N = h(id_N, t_N, c_N, r_N). \tag{17}$$

   Then, it sends $\langle tid_N, y_N, a_N, b_N, t_N \rangle$ to the *UN*.
2. On receiving the request, the *UN* first checks if the timestamp is still valid. Then, it computes:

$$k_N^* = k_{UN} \oplus a_N \oplus b_N, \tag{18}$$
$$x_N^* = h(k_{UN} \oplus k_N^*), id_N^* = x_N^* \oplus a_N, \tag{19}$$
$$r_N^* = x_N^* \oplus y_N, c_N^* = h(id_N^*, k_{UN}). \tag{20}$$

   Next, it validates tid_N by $h(id_N^*, t_N, c_N^*, r_N^*)$. The protocol will be aborted if this does not hold.
3. The *UN* continues the protocols by selecting random numbers f_N, k_N^+ and computing the following:

$$a_N^+ = id_N^* \oplus h(k_{UN}, k_N^+), \tag{21}$$
$$b_N^+ = a_N^+ \oplus k_{UN} \oplus k_N^+, \tag{22}$$
$$\eta = h(f_N, c_N^*) \oplus a_N^+, \tag{23}$$
$$\mu = h(c_N^*, f_N) \oplus b_N^+, \tag{24}$$
$$\alpha = c_N^* \oplus f_N, \tag{25}$$
$$\beta = h(id_N^*, r_N^*, f_N, \eta, \mu), \tag{26}$$
$$k_s = h(id_N^*, r_N^*, f_N, x_N^*), \tag{27}$$

   where k_s represents the session key. Finally, the *UN* sends $\langle \alpha, \beta, \eta, \mu \rangle$ to the *SN*.
4. The *SN* validates the message by computing $f_N^* = c_N \oplus \alpha$ and checking whether β equals to $h(id_N^*, r_N, f_N^*, \eta, \mu)$. If not, it rejects the protocol.
5. Finally, the *SN* computes the session keys and updates its keys, as

$$a_N^+ = h(f_N^*, c_N) \oplus \eta, \tag{28}$$
$$b_N^+ = h(c_N, f_N^*) \oplus \mu, \tag{29}$$
$$k_s^* = h(id_N, r_N, f_N^*, x_N). \tag{30}$$

   The *SN* will compute the same session key k_s as the *UN* in the absence of the adversary or noise. It will then replace (a_N, b_N) with (a_N^+, b_N^+) in its memory.

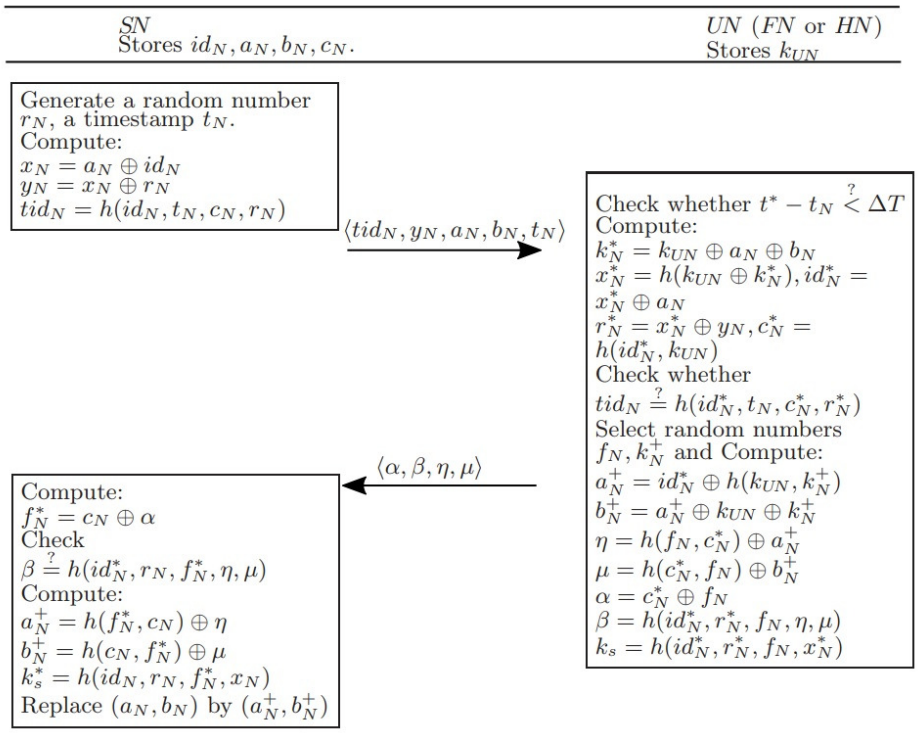

Figure 4. The repaired protocol.

5. Security Analysis of the Repaired Protocol

This section demonstrates that our repaired protocol is secure against the aforementioned attacks.

5.1. Intruding on the FN Attacks

In the repaired protocol, the *FN* no longer stores the key between an *SN* and an *HN*. Therefore, compromising an *FN* would only leak the keys between the *SN*s and the *FN*. The compromised *FN* would not be able to impersonate an *SN* to communicate with the *HN*. It is true that the compromised *FN* will still be able to access the *SN* in an *SN*-*FN* communication, but no extra access, e.g., data exclusive for the *HN*, will be given to the *FN*. This protocol also assures a secure *SN*-*FN* communication, and vice versa if all secrets stored in the *HN* are compromised.

5.2. Impersonation, Man-in-the-Middle and Replay Attacks

The protocol provides a sound mutual authentication between an *SN* and an *FN*/*HN*. The adversary defined in Section 3.1 models the necessary capabilities that requires performing impersonation, man-in-the-middle, and replay attacks. The goals of this adversary are as follows: (Goal 1) Convincing either an *SN* or a *UN* to misbelieve that a legitimate partner is participating in a communication within the timeout period; (Goal 2) Having better strategy than the wild guess in distinguishing a session key k_s against a random string with the same length. We show that there is no adversary to effectively, and with non-negligible probability, achieve either of these goals.

Goal 1 happens when either *UN* accepts or *SN* accepts. We separately discuss these cases.

- The *UN* accepts. This happens if and only if $tid_N = h(id_N^*, t_N, c_N^*, r_N^*)$. We assume that the *SN* does not generate a tid_N after $t^* - \Delta T$, otherwise it violates definition of Goal 1. If this equation is true but the hash $h(id_N^*, t_N, c_N^*, r_N^*)$ has never been computed, this will happen only with $p = 2^{-\text{length}(h)}$.

If this equation is true and the hash has been computed before, we may conclude that it is not produced by a legitimate *SN* and *UN*. This is due to the fact that id_N is unique and *SN* does not produce any at t_N and *UN* would never send computed tid_N. Therefore, the only possibility is that the adversary computes the hash by itself. This happens only if the adversary has id_N and c_N which are not sent over the network. This is bounded by $p^2 \times q_h$ where q_h is the maximum number of the hashes that are able to query with reasonable resources.

- The *SN* accepts. This happens if and only if the value of the β is equal to $h(x_N^+, r_N, f_N^*, \eta, \mu)$. Similarly, if the hash was never computed, the probability is bounded by p. If the hash is previously computed by the *UN*, the same *SN* (with id_N^*) has already sent a login request with r_N^*. Since r_N^* is randomly chosen, this happens only with $p \times q_E$, where q_E is the total number of the sessions executed by the *SN*. Otherwise, the adversary should correctly guess id_N^* and c_N, which happen only with $p^2 \times q_h$.

To sum up, the occurrence of Goal 1 has a probability lower than $(q_E + 2)p + 2q_h p^2$, where $p = 2^{-\text{length}(h)}$, q_E is the total number of the sessions executed by the *SN*, and q_h is the total number of the hashes that are able to be computed by the adversary with reasonable resources. This number is negligible when the length of the hash is large.

Goal 2 happens only when the *UN* accepts and the hash $h(id_N, r_N, f_N, x_N)$ has been computed by the adversary since k_s is never transmitted. However, id_N and x_N are both secret. A correct guess of this variable is bounded by $p^2 \times q_h$.

Considering the probability to concurrently achieve the both Goals 1 and 2, an attacker may cast as an impersonation attack, a man-in-the-middle attack, or a replay attack has a probability less than $(q_E + 2)p + 3q_h p^2$.

5.3. Tracking Attacks and Anonymity

We may see that the tracking attack, mentioned in Section 3.3, no longer operates. First of all, an *FN* serves only as a relay to replay a message. No information can be harvested to identify the relay *FN*. Furthermore, the equality $a_2 \oplus b_2 = \eta_1 \oplus \mu_1$ no longer holds, where $\eta_1 \oplus \mu_1 = a_2 \oplus b_2 \oplus h(f_N, c_N) \oplus h(c_N, f_N)$. Since c_N and f_N are not computable by the adversary, computing $h(f_N, c_N)$ or $h(c_N, f_N)$ is not possible.

6. Simulation Verification Using a Proverif Tool

Proverif is an automatic cryptographic protocol verifier, which is widely used to specify and analyze the security of authenticated key agreement protocols [19–23].

In this section, we utilize Proverif to further analyze the security and validity of the proposed protocol. In this simulation, two main roles, *SN* and *UN*, are included. The whole simulation contains the following procedures:

- First, we need to define some variables used in this simulation. K_{UN} is the secret key H_N, and SK_{SN} and SK_{UN} are the final shared key established by *SN* and *UN*, respectively—then comes the functions and events (Figure 5),
- Second, we list the goals of this simulation. More specifically, our goals is to ensure that the whole authentication process is successful, the shared key can be established, and the attacker cannot obtain the key anyway (Figure 6),
- The process of *SN* (Figure 7),
- The process of *UN* (Figure 8),
- The main execution (Figure 9).
- According to the simulation results depicted in Figure 10, we can observe that the proposed protocol can achieve the goals mentioned in Figure 6.

```
(* channel*)
free ch:channel. (* public channel *)

(* shared keys *)
free SK_SN:bitstring [private].
free SK_UN:bitstring [private].

(* constants *)
free k_UN:bitstring [private]. (* the UN's secret key *)

(* functions & reductions & equations *)
fun h(bitstring):bitstring. (* hash function*)
fun senc(bitstring,bitstring):bitstring. (* symmetric encryption *)
reduc forall m:bitstring, key:bitstring; sdec(senc(m,key),key)=m.
fun con(bitstring,bitstring):bitstring. (* concatenation operation *)
fun xor(bitstring,bitstring):bitstring. (* XOR operation*)
equation forall m:bitstring, n:bitstring;xor(xor(m,n),n)=m.

(* event *)
event BeginAuth(bitstring).
event EndAuth(bitstring).
```

Figure 5. Proverif code of variables, functions and events.

```
(* queries *)
query attacker(SK_SN).
query attacker(SK_UN).
query id:bitstring; inj-event(BeginAuth(id)) ==> inj-event(EndAuth(id)).
```

Figure 6. Goal of this simulation.

```
(* SN's process *)
let processSN(idN:bitstring,aN:bitstring,bN:bitstring,cN:bitstring) =
   new rN:bitstring;
   new tN:bitstring;
   let xN = xor(idN,aN) in
   let yN = xor(rN,xN) in
   let tidN = h(con(xor(xor(idN,tN),cN),rN)) in
   out(ch,(tidN,yN,aN,bN,tN));

   in(ch,(alpha:bitstring,beta:bitstring,eta:bitstring,mu:bitstring));
   let fN_star = xor(alpha,cN) in
   let beta_star = h( con(con(con(con(idN,rN),fN_star),eta),mu) ) in
   if beta_star = beta then
   let aN_plus = xor(eta,h(con(fN_star,cN))) in
   let bN_plus = xor(mu,h(con(cN,fN_star))) in
   let ks_star = h( con(con(con(idN,rN),fN_star),xN) ) in

   out(ch,senc(SK_SN,ks_star));
   event EndAuth(idN).
```

Figure 7. Proverif code of *SN*.

```
(* UN's process *)
let processUN =
   in(ch,(tidN:bitstring,yN:bitstring,aN:bitstring,bN:bitstring,tN:bitstring));

   let kN_star = xor(xor(bN,aN),k_UN) in
   let xN_star = h(con(k_UN,kN_star)) in
   let idN_star = xor(xN_star,aN) in
   let rN_star = xor(yN,xN_star) in
   let cN_star = h(con(idN_star,k_UN)) in
   let tidN_star = h( con(xor(xor(idN_star,tN),cN_star),rN_star) ) in
   if tidN_star = tidN then
   (
      event BeginAuth(idN_star);
      new fN:bitstring;
      new kN_plus:bitstring;
      let aN_plus = xor(h(con(k_UN,kN_plus)),idN_star) in
      let bN_plus = xor(xor(kN_plus,k_UN),aN_plus) in
      let eta = xor(h(con(fN,cN_star)),aN_plus) in
      let mu = xor(h(con(cN_star,fN)),bN_plus) in
      let alpha = xor(cN_star,fN) in
      let beta = h( con(con(con(con(idN_star,rN_star),fN),eta),mu) )    in
      out(ch,(alpha,beta,eta,mu));

      let ks = h( con(con(con(idN_star,rN_star),fN),xN_star) ) in
      out(ch,senc(SK_UN,ks))
).
```

Figure 8. Proverif code of *HN*.

```
(* ----- Main ----- *)
process
   (* register a new sensor node *)
   new idN:bitstring;
   new kN:bitstring;
   let aN = xor(h(con(k_UN,kN)),idN) in
   let bN = xor(xor(kN,k_UN),aN) in
   let cN = h(con(idN,k_UN)) in
   ( !processSN(idN,aN,bN,cN) ) | ( !processUN )
```

Figure 9. Main process of this simulation.

```
-- Query inj-event(BeginAuth(id)) ==> inj-event(EndAuth(id))
RESULT inj-event(BeginAuth(id)) ==> inj-event(EndAuth(id)) is true.

-- Query not attacker(SK_UN[])
RESULT not attacker(SK_UN[]) is true.

-- Query not attacker(SK_SN[])
RESULT not attacker(SK_SN[]) is true.
```

Figure 10. Simulation results.

7. Performance Evaluation

This section describes performance evaluation of the repaired protocol along with other related protocols [4,10–12,14] in security properties and estimated time. We focus on the security against the anonymity, tracking attack, insider attack, replay attack, impersonation attack, man-in-the-middle attack, mutual authentication and the session key forward secrecy. From Table 1, we see that only the repaired protocol, Wu's protocol [12] and Shen et al. [14] fulfill all the security properties.

Table 1. Comparison of the security properties. Y and N stands for fulfilling and not fulfilling the requirement respectively.

	C1	C2	C3	C4	C5	C6	C7	C8
[10]	Y	Y	N	Y	Y	Y	Y	Y
[11]	N	N	Y	Y	Y	Y	Y	Y
[12]	Y	Y	Y	Y	Y	Y	Y	Y
[4]	N	N	N	Y	N	Y	Y	Y
[14]	Y	Y	Y	Y	Y	Y	Y	Y
Ours	Y	Y	Y	Y	Y	Y	Y	Y

C1: Provide anonymity;
C2: Withstand tracking attack;
C3: Withstand insider attack;
C4: Withstand repay attack;
C5: Withstand impersonation attack;
C6: Withstand man-in-the-middle attack;
C7: Mutual authentication;
C8: The session key forward secrecy

We analyze the time performance of these protocol by analysis of the core cryptographic operations used in each of them, and then estimate the running time of these protocols by adding the time of executed cryptographic operations. We do not consider the possibility of parallel computation with multi-core technologies since most wearable devices are only single core. Pipelining is also not discussed here since the authentication usually needs to be executed once.

We consider two possible realizations of an *SN*. A sensor device using the MICAz with 4 KB RAM (Crossbow Technology, San Jose, CA, USA) and 7-MHz ATmega128L microcontroller (Microchip Technology Inc, Chandler, AZ, USA) and a smart phone using an iPhone 6s (Apple, Cupertino, CA, USA) with 2 GB RAM ARM (armv8-a) CPU. The data are taken from [13,24,25] for the time required on the MICAz while we implement those implementations on a smart phone using the Pairing Based Cryptographic Library [26]. The result is summarized in Table 2.

Table 2. Computation of the cryptographic operations.

Symbol	Description	Running Time on a Smartphone	Running Time on a MICAz
T_h	Hash function	0.03 ms	8 ms [25]
T_{sym}	Symmetric encryption/description operation	0.12 ms	3.5 ms [24]
T_{sm}	Scalar multiplication over elliptic curves	20.23 ms	2450 ms [13]
T_{bp}	Bilinear pairing operation	25.64 ms	5320 ms [13]

Table 3 lists the estimated time of the mentioned protocols, considering the above experimental data. From this table, we may observe that the repaired protocol costs more time than Li's protocol [4] as it takes six more hash functions, but costs less time than the other related protocols [10–12,14] .

Table 3. Comparison of the estimated time.

Protocols	Time Cost	Running Time on a Smartphone	Running Time on a MICAz
[10]	$4T_h + 5T_{sm} + 3T_{bp}$	178.19 ms	28242 ms
[11]	$11T_h + 9T_{sm} + 3T_{sym}$	182.64 ms	22148.5 ms
[12]	$7T_h + 8T_{sm} + T_{bp} + 2T_{sym}$	187.93 ms	24983 ms
[4]	$9T_h$	0.27 ms	72 ms
[14]	$9T_h + 13T_{sm}$	263.26 ms	31922 ms
Ours	$15T_h$	0.45 ms	120 ms

8. Conclusions

We demonstrated that Li's protocol is broken and should not be used in any application implementation related to the WBAN. At the same time, we proposed another architecture that research should be considered when designing any authentication. In this architecture, the linear relationship connecting an *SN* to an *FN* and an *FN* to an *HN* is abandoned. Instead, *SN* s, *FN* s and *HN* s are directly connected to each other through a pairwise secret. The *FN* changes its role in an *SN*-*HN* communication from coordinating to relaying messages between the *SN* and *HN*. We believe that this approach is highly effective and secure so that compromise of the *HN* or *FN* would not lead to a total compromise of the system. In such an architecture, an *FN* may be abused through consuming the relay service by attackers. This problem, however, appears in most of the relaying systems in all wireless networks, which may be handled via some firewall rules or intrusion detection techniques. This represents an interesting research topic to be further studied by the authors in the future.

Author Contributions: C.-M.C. and K.-H.W. wrote the main concepts of the manuscript; B.X. designed and implemented the experiments; T.-Y.W. checked the English writing and organization of the manuscript.

Funding: The work of Chien-Ming Chen was supported in part by Shenzhen Technical Project under Grant number JCYJ20170307151750788 and in part by Shenzhen Technical Project under Grant number KQJSCX20170327161755. The work of Tsu-Yang Wu was supported in part by the Science and Technology Development Center, Ministry of Education, China under Grant no. 2017A13025 and the Natural Science Foundation of Fujian Province under Grant no. 2018J01636.

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

References

1. Dussault, C.; Toeg, H.; Nathan, M.; Wang, Z.J.; Roux, J.F.; Secemsky, E. Electrocardiographic Monitoring for Detecting Atrial Fibrillation After Ischemic Stroke or Transient Ischemic Attack. *Circ. Arrhythm. Electrophysiol.* **2015**, *8*, 263–269. [CrossRef] [PubMed]
2. Epstein, L.J.; Kristo, D.; Strollo, P.J.; Friedman, N.; Malhotra, A.; Patil, S.P.; Ramar, K.; Rogers, R.; Schwab, R.J.; Weaver, E.M.; et al. Clinical guideline for the evaluation, management and long-term care of obstructive sleep apnea in adults. *J. Clin. Sleep Med.* **2009**, *5*, 263–276. [PubMed]
3. Toorani, M. On Vulnerabilities of the Security Association in the IEEE 802.15.6 Standard. In Proceedings of the Financial Cryptography and Data Security: FC 2015 International Workshops, BITCOIN, WAHC, and Wearable, San Juan, Puerto Rico, 26–30 January 2015; pp. 245–260.
4. Li, X.; Ibrahim, M.H.; Kumari, S.; Sangaiah, A.K.; Gupta, V.; Choo, K.K.R. Anonymous mutual authentication and key agreement scheme for wearable sensors in wireless body area networks. *Comput. Netw.* **2017**, *129*, 429–443. [CrossRef]
5. Kaufman, C.; Hoffman, P.; Nir, Y.; Eronen, P. *Internet Key Exchange Protocol Version 2 IKEv2*; RFC 5996, RFC Editor; IETF: Fremont, CA, USA, 2010.
6. Wang, K.H.; Chen, C.M.; Fang, W.; Wu, T.Y. On the security of a new ultra-lightweight authentication protocol in IoT environment for RFID tags. *J. Supercomput.* **2017**, *74*, 1–6. [CrossRef]
7. Wang, D.; Wang, P. On the anonymity of two-factor authentication schemes for wireless sensor networks: Attacks, principle and solutions. *Comput. Netw.* **2014**, *73*, 41–57. [CrossRef]

8. Keoh, S.L.; Lupu, E.; Sloman, M. Securing body sensor networks: Sensor association and key management. In Proceedings of the 2009 IEEE International Conference on Pervasive Computing and Communications, PerCom 2009, Galveston, TX, USA, 9–13 March 2009; pp. 1–6.
9. Liu, J.; Kwak, K.S. Hybrid security mechanisms for wireless body area networks. In Proceedings of the 2010 Second International Conference on Ubiquitous and Future Networks (ICUFN), Jeju, Korea, 16–18 June 2010; pp. 98–103.
10. Liu, J.; Zhang, Z.; Chen, X.; Kwak, K.S. Certificateless remote anonymous authentication schemes for wirelessbody area networks. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 332–342. [CrossRef]
11. Zhao, Z. An efficient anonymous authentication scheme for wireless body area networks using elliptic curve cryptosystem. *J. Med. Syst.* **2014**, *38*, 13. [CrossRef] [PubMed]
12. Wu, L.; Zhang, Y.; Li, L.; Shen, J. Efficient and anonymous authentication scheme for wireless body area networks. *J. Med. Syst.* **2016**, *40*, 134. [CrossRef] [PubMed]
13. Xiong, X.; Wong, D.S.; Deng, X. TinyPairing: A Fast and Lightweight Pairing-Based Cryptographic Library for Wireless Sensor Networks. In Proceedings of the 2010 IEEE Wireless Communication and Networking Conference, Sydney, NSW, Australia, 18–21 April 2010; pp. 1–6.
14. Shen, J.; Gui, Z.; Ji, S.; Shen, J.; Tan, H.; Tang, Y. Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. *J. Netw. Comput. Appl.* **2018**, *106*, 117–123. [CrossRef]
15. Venkatasubramanian, K.K.; Banerjee, A.; Gupta, S.K.S. PSKA: Usable and secure key agreement scheme for body area networks. *IEEE Trans. Inf. Technol. Biomed.* **2010**, *14*, 60–68. [CrossRef] [PubMed]
16. Zhang, Z.; Wang, H.; Vasilakos, A.V.; Fang, H. ECG-cryptography and authentication in body area networks. *IEEE Trans. Inf. Technol. Biomed.* **2012**, *16*, 1070–1078. [CrossRef] [PubMed]
17. Shi, L.; Yuan, J.; Yu, S.; Li, M. ASK-BAN: Authenticated secret key extraction utilizing channel characteristics for body area networks. In Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks, Budapest, Hungary, 17–19 April 2013; ACM: New York, NY, USA, 2013; pp. 155–166.
18. Wang, K.H.; Chen, C.M.; Fang, W.; Wu, T.Y. A secure authentication scheme for Internet of Things. *Pervasive Mob. Comput.* **2017**, *42*, 15–26. [CrossRef]
19. Jiang, Q.; Zeadally, S.; Ma, J.; He, D. Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks. *IEEE Access* **2017**, *5*, 3376–3392. [CrossRef]
20. Chaudhry, S.A.; Naqvi, H.; Sher, M.; Farash, M.S.; Hassan, M.U. An improved and provably secure privacy preserving authentication protocol for SIP. *Peer-to-Peer Netw. Appl.* **2017**, *10*, 1–15. [CrossRef]
21. Wu, F.; Xu, L.; Kumari, S.; Li, X.; Shen, J.; Choo, K.K.R.; Wazid, M.; Das, A.K. An efficient authentication and key agreement scheme for multi-gateway wireless sensor networks in IoT deployment. *J. Netw. Comput. Appl.* **2017**, *89*, 72–85. [CrossRef]
22. Abbasinezhad-Mood, D.; Nikooghadam, M. Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended chebyshev chaotic maps. *IEEE Trans. Ind. Inform.* **2018**. [CrossRef]
23. Abbasinezhad-Mood, D.; Nikooghadam, M. Design and hardware implementation of a security-enhanced elliptic curve cryptography based lightweight authentication scheme for smart grid communications. *Future Gener. Comput. Syst.* **2018**, *84*, 47–57. [CrossRef]
24. Panait, C.; Dragomir, D. Measuring the performance and energy consumption of AES in wireless sensor networks. In Proceedings of the 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), Lodz, Poland, 13–16 September 2015; pp. 1261–1266.
25. Koschuch, M.; Hudler, M.; Saffer, Z. Towards algorithm agility for wireless sensor networks: Comparison of the portability of selected Hash functions. In Proceedings of the 2013 International Conference on Data Communication Networking (DCNET), Reykjavik, Iceland, 29–31 July 2013; pp. 1–5.
26. Lynn, B. On the Implementation of Pairing-Based Cryptosystems. Ph.D. Thesis, Stanford University Stanford, Stanford, CA, USA, 2007.