Hindawi
Security and Communication Networks
Volume 2021, Article ID 5598574, 12 pages
https://doi.org/10.1155/2021/5598574

*Research Article*

# Privacy Enhancement on Unilateral Bluetooth Authentication Protocol for Mobile Crowdsensing

**Da-Zhi Sun**[1] **and Ji-Dong Zhong**[2]

[1]*Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, No. 135, Yaguan Road, Tianjin Haihe Education Park, Tianjin 300350, China*
[2]*Department of Computer Science and Engineering, Tongji University, No. 4800, Caoan Road, Jiading District, Shanghai 201804, China*

Correspondence should be addressed to Da-Zhi Sun; sundazhi@tju.edu.cn

Received 15 February 2021; Accepted 5 May 2021; Published 8 June 2021

Academic Editor: Ximeng Liu



As an open standard for the short-range radio frequency communications, Bluetooth is suitable for Mobile Crowdsensing Systems (MCS). However, the massive deployment of personal Bluetooth-enabled devices also raises privacy concerns on their wielders. Hence, we investigate the privacy of the unilateral authentication protocol according to the recent Bluetooth standard v5.2. The contributions of the paper are twofold. (1) We demonstrate that the unilateral authentication protocol suffers from privacy weakness. That is, the attacker is able to identify the target Bluetooth-enabled device once he observed the device's previous transmitted messages during the protocol run. More importantly, we analyze the privacy threat of the Bluetooth MCS, when the attacker exploits the proposed privacy weakness under the typical Internet of Things (IoT) scenarios. (2) An improved unilateral authentication protocol is therefore devised to repair the weakness. Under our formal privacy model, the improved protocol provably solves the traceability problem of the original protocol in the Bluetooth standard. Additionally, the improved protocol can be easily adapted to the Bluetooth standards because it merely employs the basic cryptographic components available in the standard specifications. In addition, we also suggest and evaluate two countermeasures, which do not need to modify the original protocol.

## 1. Introduction

Bluetooth [1] is an open technology standard for wireless short-range radio frequency communications. Bluetooth hardware and software modules are already integrated into many kinds of consumer and business devices including smart phones, headsets, laptops, keyboards, mouses, tablets, and automobiles. Actually, Bluetooth offers a highly practical approach to establishing Mobile Crowdsensing Systems (MCS) because of its universality, convenience, and adaptation.

In order to protect device users and their sensitive data, Bluetooth provides a security solution for the hostile environments [2]. More precisely, the effective Bluetooth standard specifications [3–5] define four security modes, namely, modes 1 through 4. Each Bluetooth-enabled device must operate in one of the four security modes. Security mode 1 does not employ any security measure. Security modes 2 and 4 are treated as the service level-enforced security modes. That is to say, the device in security mode 2 or 4 will not start any security procedure until it receives or initiates a channel establishment request. Security mode 3 is the link level-enforced security mode, where the security procedures are initiated before the physical link is fully established. Security modes 2, 3, and 4 are all composed of three crucial procedures, i.e., pairing and link key generation, authentication, and confidentiality. It needs to be pointed out that the authentication procedure and the confidentiality procedure in security modes 2, 3, and 4 are fully the same. We briefly review the three procedures as follows:

Pairing and link key generation: this procedure is responsible for establishing the link key between a pair of the devices and further binding the trusted devices. The link key will be used throughout the subsequent security procedures. Security modes 2 and 3 employ the same pairing and link key generation scheme called Personal Identification Number (PIN) pairing. The PIN pairing is analyzed and improved in the literature such as [6, 7]. However, in security mode 4, Secure Simple Pairing (SSP) is used instead. A series of works such as [8–11] address the security properties of SSP under the distinctive practical concerns.

Authentication: this procedure applies to two paired devices and needs their shared link key generated by the previous procedure. The procedure goal is to validate the legitimate identity claimed by the pairing device itself. When the authentication attempt fails, any retry with the same identity will be delayed for a waiting interval.

Confidentiality: this procedure provides a separate confidential service to the data transmitted between the pairing devices.

The widespread use of Bluetooth-enabled devices has given birth to many wireless personal applications, such as connecting mobile phones to wireless headsets, emergency systems of cars, and digital wallets and merchants. Bluetooth is often used for establishing Wireless Personal Area Network (WPAN) because of its usability and performance. In fact, the Bluetooth standard is adapted in IEEE 802.15 [12] for WPAN. Obviously, it is important to protect the privacy of the users under Bluetooth WPAN. However, the absence of physical contact during communications and the expected ubiquity of sensitive applications (such as the MCS [13, 14]) will encourage nefarious entities to observe and track the devices through their transmitted Bluetooth messages. In a Bluetooth WPAN application, the attacker may intercept and analyze the transmitted messages among devices. If at this time a device is linked to a user, the identity of the user will then be disclosed by his device. Figure 1 illustrates a Bluetooth application scenario where the attacker intercepts and analyzes the transmitted messages among devices. Hence, the need for Bluetooth to be resistant against the privacy threats arises. To guarantee the privacy of the user, the transmitted messages of the security mode in particular should not be exploited to identify the target device.

The National Institute of Standards and Technology (NIST) [15] surveyed the privacy features and threats according to the early versions of the Bluetooth standard. A few of works [16–18] showed that the privacy of a particular user can be compromised if the Bluetooth-enabled device address associated with the user is captured, and therefore proposed the improved schemes to repair it. Moreover, some devices [19, 20] have been implemented with the protection mechanisms for their Bluetooth addresses. Many researchers [21–25] found that the public advertising channels in Bluetooth Low Energy (BLE) may leak the identity information of the device and therefore designed the privacy countermeasures. Celosia and Cunche [26] reported a timing attack, which can be triggered by a remote attacker, to infer the state of a device from Bluetooth information and undermine the privacy of the user. As a potential privacy threat, Bluetooth traffic can be sniffed even if the device is in indiscoverable mode, as demonstrated by Albazrqaoe et al. [27]. Bello-Ogunu et al. [28] developed a privacy management framework that provides a policy configuration platform for BLE beacon. We [29] addressed the privacy vulnerability of SSP's BLE version in security mode 4. In addition, Bluetooth systems [30–32] in the application level are designed to protect the privacy of the user. Due to the fast development of Bluetooth WPAN and its integration into Internet of Things (IoT), more and more privacy protection features are now already included in the Bluetooth standard specifications [4].

To the best of our knowledge, there is still no active research on the privacy of the authentication procedure in the Bluetooth security solution. However, the attacker may exploit the vulnerabilities in the authentication procedure to compromise the privacy of the user, especially when the Bluetooth-enabled devices are deployed in the IoT environment. Moreover, the overall strength of privacy protection would be dominated by the weakest procedure in the Bluetooth security solution. Hence, we focus on the privacy enhancement of the authentication procedure according to the recent Bluetooth standard v5.2 [4].

Two challenge-response protocols, i.e., the unilateral (or legacy) authentication protocol and the mutual (or secure) authentication protocol, are used to realize the authentication procedure in the Bluetooth standard v5.2. In this paper, we will systematically investigate the privacy of the unilateral authentication protocol. We demonstrate that the attacker can track the target Bluetooth-enabled device once he observed the device's previous transmitted messages during the protocol run. We further evaluate its impact on the Bluetooth MCS, when our proposed privacy weakness is exploited under typical IoT scenarios. An improved unilateral authentication protocol is therefore proposed to overcome the privacy weakness in the original protocol. Without high extra implementation costs, our improved protocol provably solves the traceability problem in the original protocol. In addition, two non-protocol countermeasures are also suggested and evaluated for the privacy enhancement.

## 2. Review of Unilateral Bluetooth Authentication Protocol

In the unilateral authentication protocol [4], each Bluetooth-enabled device is referred to as either the claimant or the verifier. The claimant is a device manifesting its own identity to the verifier, and the verifier is a device validating the identity of the claimant. The protocol validates the devices by verifying the knowledge of the shared link key, which is established in the pairing and link key generation procedure. The protocol makes use of a cryptographic hash algorithm $E_1$


Figure 1: Interception of personal Bluetooth communication.

with an output of 128 bits. The algorithm $E_1$ is based on the SAFER + block cipher but with some minor modifications [4]. Let $K_{\text{LINK}}$ be the shared link key between the claimant and the verifier. Let BD_ADDR be the claimant's device address. The protocol is shown as Figure 2, and the authentication session is described as follows.

Step 1: the verifier generates a 128-bit random number AU_RAND as the challenge and then sends it to the claimant.

Step 2: both devices calculate the authentication token $\{\text{SRES, ACO}\} = E_1\ (K_{\text{LINK}}, \text{BD\_ADDR, AU\_RAND})$, where SRES known as the signed response is the 32 most significant bits of the 128-bit output of the algorithm $E_1$, and ACO known as authenticated ciphering offset is the remaining bits for creating the encryption key in the confidentiality procedure.

Step 3: the authentication response SRES is sent from the claimant to the verifier.

Step 4: the verifier compares the received SRES with the counterpart calculated locally. If they are the same, the protocol run succeeds and the verifier accepts the identity of the claimant; otherwise, the protocol run fails.

The link key is either semi-permanent or temporary according to Bluetooth standard specifications [4]. A semi-permanent link key may be still used after the current secure session is terminated. This implies that a semi-permanent link key may be activated in the authentication of several subsequent connections between the devices. The lifetime of a temporary link key is limited by the lifetime of the current secure session; that is, it shall not be reused in a later secure session. In brief, a link key will remain valid until the next successful run of the pairing and link key generation procedure.

## 3. Privacy Weakness of Unilateral Bluetooth Authentication Protocol

Assume that the attacker eavesdrops and records AU_RAND and SRES during a protocol run between the claimant (Device 1) and the verifier (Device 2). Herein, this assumption is reasonable because AU_RAND and SRES are insecurely transmitted over a Bluetooth wireless channel. As shown in Figure 3, the attacker can use the past AU_RAND and SRES to validate whether the claimant is Device 1 or not by the following steps.

Step 1: replay the recorded AU_RAND in Step 1 of the unilateral authentication protocol.

Step 2: omit Step 2 of the unilateral authentication protocol.

Step 3: upon receiving the claimant's SRES in Step 3 of the unilateral authentication protocol, compare it with the recorded SRES. If they are equal, the claimant is Device 1.

If the claimant is Device 1, the same BD_ADDR, AU_RAND, and $K_{\text{LINK}}$ should be used as the input of algorithm $E_1$ during the attacker's authentication run. Hence, the algorithm $E_1$ will output the same SRES as the attacker's recorded one. On the contrary, if the claimant is not Device 1, the different $K_{\text{LINK}}$ and BD_ADDR should be used as the input of the algorithm $E_1$. In this case, the algorithm $E_1$ will output the same SRES as the previous recorded one with a negligible probability. Hence, the attacker can always use

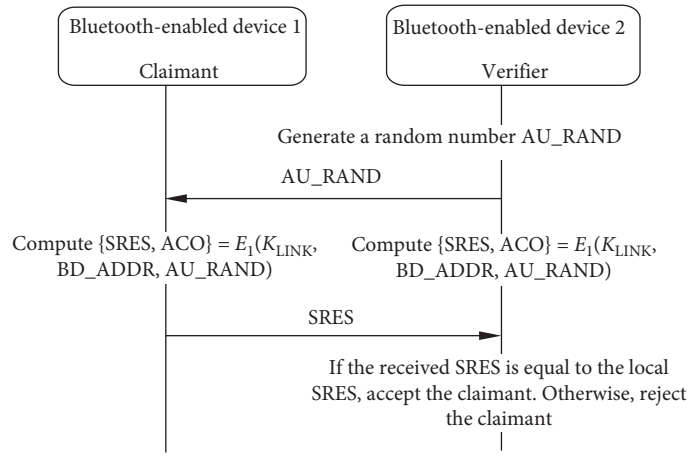


Figure 2: Bluetooth authentication protocol.

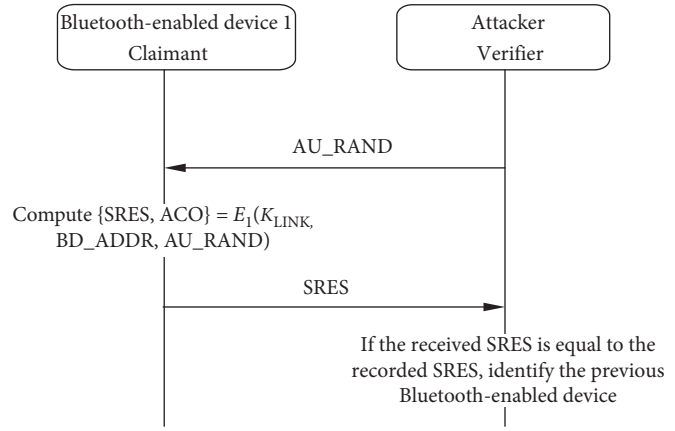


Figure 3: Bluetooth-enabled device tracing by authentication process.

the proposed attack to identify Device 1. We further discuss the proposed attack as follows.

To prevent the proposed attack, the device can update its $K_{LINK}$ before each authentication session. However, we argue that it is an impractical method due to the heavy overheads of the pairing and link key generation procedure.

If $K_{LINK}$ and BD_ADDR are available, the attacker is also able to identify the device by the authentication session. However, this implies that the attacker must be powerful enough to break into the device. In practice, it is hard for the attacker to directly crack a device.

The proposed attack merely exploits the vulnerabilities in the unilateral authentication protocol of the Bluetooth standard. Moreover, since security modes 2, 3, and 4 all employ this protocol, the proposed attack is regarded as a broad-spectrum tracking method.

In addition, it is worth noting that the Bluetooth standard specifications [4] give a countermeasure to prevent the attacker from repeating the authentication procedure. That is, "when the authentication attempt fails, a waiting interval shall pass before the verifier will initiate a new authentication attempt to the same claimant, or before it will respond to an authentication attempt initiated by a device claiming the same identity as the failed device. For each subsequent authentication failure, the waiting interval shall be increased exponentially. For example, after each failure, the waiting interval before a new attempt can be made could be twice as long as the waiting interval prior to the previous attempt. The waiting interval shall be limited to a maximum."

However, this countermeasure cannot overcome the proposed privacy weakness on the unilateral authentication protocol because the attacker merely uses one authentication attempt to confirm the identity of the claimant.

## 4. Privacy Threat Analysis of Bluetooth MCS due to Proposed Weakness

In some Bluetooth applications such as [9, 11], the privacy of the user is not a serious concern. However, when the Bluetooth devices enter into the MCS, we must take user privacy very seriously because a large amount of personal sensitive data may be collected automatically. In this section, we use the proposed privacy weakness to analyze the privacy threat of the Bluetooth MCS under several typical IoT scenarios.

We know that MCS architecture always consists of three tiers, i.e., the devices, the edge gateway, and the cloud. The devices include networked sensors, actuators, and embedded communication hardware, which adopt the widely used standards such as Bluetooth and Zigbee.

*4.1. Bluetooth IoT Scenarios.* Since the standard v4.2 [33], the Bluetooth group begins to promote the IoT technology. With the fast development of the IoT applications, many MCS designs are proposed by using Bluetooth-enabled devices and their networks. For example, the MCS architecture under the Bluetooth IoT [34] can be divided into six layers: hardware layer, microcontroller layer, Bluetooth connectivity layer, connectivity layer, Bluetooth IoT cloud stack layer, and application layer. We can enumerate several typical Bluetooth IoT scenarios as follows.

*4.1.1. Child Care.* In this scenario such as [35], children wear wristbands with Bluetooth-enabled devices, and the Bluetooth network would provide the nursery teachers with information on whether the children are still within reach. Furthermore, the nursery administrators could get statistics of the total daily children movement information from their Bluetooth-enabled devices.

*4.1.2. Medical and Healthcare.* Bluetooth-enabled health monitoring devices [36] can be deployed to realize remote health monitoring and emergency notification systems. These devices can range from blood pressure and heart rate monitors to advanced devices capable of monitoring specialized implants. Furthermore, if the hospital beds are equipped with Bluetooth-enabled devices, the doctor can detect whether the hospital bed is occupied or when a patient is attempting to get up.

*4.1.3. Animal Tracker.* By mounting the relay nodes around the pastures, farmers can monitor the livestock with Bluetooth-enabled devices and keep track of each individual animal of the herd. Moreover, with the Bluetooth data of the livestock, the precision feed mechanisms can be implemented with using artificial intelligence to count the number of the livestock, analyze the health trend of the livestock, and evaluate the breeding effectiveness.

*4.1.4. Barcode Scanner.* Bluetooth IoT is suitable for barcode scanner applications, since Bluetooth network could cover a large warehouse or even multiple large warehouses potentially with a lot of obstacles and walls. Each warehouse worker may use his own barcode scanner with the constant Bluetooth network coverage.

*4.1.5. Greenhouse Monitoring System.* Bluetooth-enabled devices could collect plant data on temperature, rainfall, humidity, wind speed, pest infestation, and soil content and further communicate with sprinkler and ventilation systems. The user could manually control and configure systems of the greenhouse with an app using smart phone as a Bluetooth controller. This greenhouse facility could easily be expanded by adding more and more nodes when necessary.

*4.1.6. Battlefield Surveillance.* Soldiers and military equipment with the wireless network access (called the node) could search the objects for probably hostile forces. Then, they provide the real-time situational awareness to the base station which in turn sends over those data to other nodes as well as to the command center. It is high time that the military may tend to use commercial off-the-shelf Bluetooth sensors due to their inherent price advantages.

*4.2. Proposed Weakness for Bluetooth MCS.* Here, we show that the proposed attack endangers the privacy of the device owner and further poses a threat to the privacy of the Bluetooth MCS.

The sophisticated attacker sends his owner AU_RANDs to the target devices and records their corresponding SRESs. He also collects the target devices' AU_RANDs and SRESs over the Bluetooth channel. These values can be stored in a data table as shown in Figure 4. The attacker may remove AU_RAND and SRES from the data table, if their corresponding $K_{\mathrm{LINK}}$ expires. He simultaneously attempts to initiate the authentication runs with all potential devices within the reach of the imitative device under his control. If any device responds to the right SRES, the corresponding identity linkage will be disclosed to the attacker. With supplementary information such as locations, times, user behavior, and known identities, the attacker can deduce the user identities of the target devices and compromise the privacy of the Bluetooth MCS.

For example, consider the child care scenario in Section 4.1. The proposed privacy weakness can be exploited to track how a victim child's Bluetooth-enabled device within some area moves. This helps the attacker to infer the child's movement characteristics. What is more important, the attacker can derive the relationship among children via a great deal of the tracked Bluetooth-enabled devices.

*4.3. Privacy Threat of Bluetooth MCS.* We now analyze the privacy threat of the MCS under the IoT scenarios in Section 4.1, when the attacker exploits the proposed privacy weakness. In Table 1, we first collect and analyze the privacy features of the MCS on three criteria: the correlation between the device and the user identity, the system deployment range, and the domain's privacy demands. Then, we can comprehensively evaluate the privacy features of the MCS and the proposed privacy weakness discussed in Section 4.2. Finally, we deduce the privacy threat level as in Table 2.

## 5. Improved Unilateral Bluetooth Authentication Protocol

In this section, we improve the unilateral authentication protocol to prevent the proposed privacy weakness. To be

| ID field | AU_RAND field | SRES field |
|---|---|---|
| $ID_1$ | $AU\_RAND_1$ | $SRES_1$ |
| $ID_2$ | $AU\_RAND_2$ | $SRES_2$ |
| ⋮ | ⋮ | ⋮ |

Figure 4: Multiple-device tracking table.

compatible with the standard, the improvement should be built by the same cryptographic components as in the original unilateral authentication protocol.

*5.1. Protocol Description.* As shown in Figure 5, we propose an improved protocol to repair the traceability weakness in the original protocol. The authentication session is now as follows.

Step 1: the verifier generates a random number $AU\_RAND_V$ as the challenge and then sends it to the claimant.

Step 2: the claimant creates another random number $AU\_RAND_C$ and calculates the authentication token $\{SRES, ACO\} = E_1\ (K_{LINK}, BD\_ADDR, AU\_RAND_V, AU\_RAND_C)$. Herein, based on the SAFER + block cipher, it is necessary for the cryptographic hash algorithm $E_1$ to have an input with a larger size. Alternatively, both $AU\_RAND_V$ and $AU\_RAND_C$ can be set to 64 bits, and the algorithm $E_1$ remains the same.

Step 3: the claimant sends the response $AU\_RAND_C$ and SRES to the verifier.

Step 4: upon receiving $AU\_RAND_C$ and SRES, the verifier first uses $AU\_RAND_C$ to calculate the authentication token $\{SRES, ACO\} = E_1\ (K_{LINK}, BD\_ADDR, AU\_RAND_V, AU\_RAND_C)$. Then, it compares the received SRES with the counterpart calculated locally. If they are the same, the protocol run is successful and the verifier accepts the identity of the claimant; otherwise, the protocol run fails.

*Remark 1*

In practice, we can employ a non-symmetric split strategy for the random numbers $AU\_RAND_V$ and $AU\_RAND_C$. If the risk of the tracking devices is low in some applications, the claimant's $AU\_RAND_C$ can be shortened to 32 bits or about, while keeping the length of the verifier's $AU\_RAND_V$ longer to ensure that the actual authentication strength is not degraded.

The random number $AU\_RAND_C$ can also be replaced with a sequence number, which never repeats in each protocol run. One example of the sequence number is a counter. It demands that the state information of the claimant be maintained after a protocol run. However, the bit length of the sequence number can be very short.

*5.2. Efficiency Comparison.* In this section, we compare the implementation costs of the improved protocol and the original one. Clearly, both protocols have the same secret storage cost due to the same $K_{LINK}$. As far as the computation cost is concerned, both protocols need one cryptographic hash computation to obtain {SRES, ACO} in each device. For the communication cost, the improved protocol has to transmit $AU\_RAND_V$, $AU\_RAND_C$, and SRES, whereas the original protocol only transmits AU_RAND and SRES. When $AU\_RAND_V$ and $AU\_RAND_C$ both are 128 bits, the improved protocol incurs an additional overhead of transmitting 128 bits in an authentication session. However, such overhead is insignificant in most of the devices. When $AU\_RAND_V$ and $AU\_RAND_C$ both are 64 bits, the communication cost of both protocols is the same. In conclusion, the improved protocol is as efficient as the original protocol.

## 6. Privacy Evaluation of Improved Unilateral Bluetooth Authentication Protocol

A fact we shall see from the proposed privacy weakness in Section 3 is that the design of the unilateral authentication protocol is extremely error-prone, even though it originated in standard documents. Therefore, to avoid the design defects as much as possible, we adopt the formal method to examine the privacy of the improved unilateral authentication protocol in the following.

Let $\{0, 1\}^*$ denote the set of finite binary strings and $\{0, 1\}^l$ represent the set of binary strings of the bit length $l$. Let Pr[Ev] be the probability of the event Ev and $Pr[Ev_1|Ev_2]$ the conditional probability of the event $Ev_1$ with respect to the event $Ev_2$.

*6.1. Model Definition.* We use a formal model to evaluate the privacy of the unilateral authentication protocols. Similar work to improve the privacy analysis of the authentication protocols begins with the paper of Juels and Weis [37]. Under the Bluetooth network setting, let $I = \{1, 2, \ldots, n\}$ be a set of Bluetooth-enabled devices with either a claimant or a verifier. The unilateral authentication protocol $\Pi$ rules how the claimant and the verifier behave during the protocol run. For any $i, j \in I$, let $\Pi_{i,j}$ be $i$'s instance of $\Pi$ interacting with $j$. According to $\Pi$, $\Pi_{i,j}$ generates, transmits, and receives the message(s) to authenticate the claimant. To some extent, $\Pi_{i,j}$ can be treated as an efficiently computable function. The internal state of $\Pi_{i,j}$ includes the following variables:

sid: the unique identifier of a protocol run.

$K_{LINK}$: the link key shared by both $i$ and $j$.

BD_ADDR: the claimant address.

tran: a transcript of $i$'s current protocol run so far, i.e., the ordered set of messages transmitted and received by $i$ so far.

$\delta$: a Boolean variable set to true or false denoting whether to accept or reject at the end of the protocol run. This variable is merely valid in the verifier's instance.

Without loss of generality, we assume that $i$ is the verifier and $j$ is the intended claimant in the following. A protocol run can be modeled by collaboratively running $\Pi_{i,j}$ and $\Pi_{j,i}$. At the end of the protocol run, $i$ should either accept or reject

Table 1: Privacy features of Bluetooth IoT scenarios.

| Feature/scenario | Child care | Medical and healthcare | Animal tracker | Barcode scanner | Greenhouse monitoring system | Battlefield surveillance |
|---|---|---|---|---|---|---|
| Systems based on device implants | No | Possible | Possible | No | Possible | Possible |
| Device associated with a person | Yes | Possible | No | Possible | No | Possible |
| Fixed link between device and identity | Yes | Yes | Yes | Possible | Possible | Possible |
| System operating locally within a restricted area | Yes | Yes | No | Yes | No | No |
| System within a single organization | No | No | Yes | No | No | Yes |
| System operating across different organizations | No | No | No | No | Yes | No |
| Privacy demand | High | High | Low | Medium | Low | High |

Table 2: Privacy threat incurring by proposed weakness.

| | Child care | Medical and healthcare | Animal tracker | Barcode scanner | Greenhouse monitoring system | Battlefield surveillance |
|---|---|---|---|---|---|---|
| Privacy threat level | High | High | Low | Medium | Low | High |

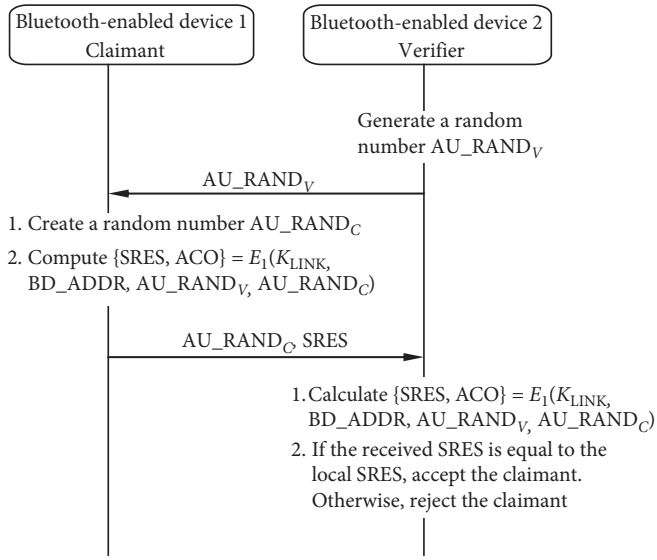


Figure 5: Improved Bluetooth unilateral authentication protocol.

the purported identity of $j$, which is indicated by $\Pi_{i,j}$'s $\delta$. Here, $i$ verifies $j$ by using $\Pi_{i,j}$'s $K_{\text{LINK}}$ and BD_ADDR.

A function $\varepsilon$: $\mathbf{N} \longrightarrow \mathbf{R}$ is negligible in $n$ if for all constants $c \geq 0$ there always exists an integer $N$ such that for all integers $n > N$ it holds that $\varepsilon(n) < n^{-c}$. If $\varepsilon$ is negligible, then $1 - \varepsilon$ is said to be an overwhelming. Let $k$ or $1^k$ be the security parameter of the unilateral authentication protocol $\Pi$. We firstly define the notion of correctness as follows.

*Definition 1* (correctness). A unilateral authentication protocol $\Pi$ with the security parameter $k$ is correct if, given any honest verifier $i \in I$ and any honest claimant $j \in I$, the protocol run of the pair $\Pi_{i,j}$ and $\Pi_{j,i}$ succeeds with overwhelming probability in $k$.

Correctness of $\Pi$ means that if both $\Pi_{i,j}$ and $\Pi_{j,i}$ collaboratively generate tran using the same $K_{\text{LINK}}$ and BD_ADDR, $\Pi_{i,j}$'s $\delta$ is true at the end of the protocol run. Clearly, each $\Pi$ must satisfy it. It is easy to check that both original and improved protocols are correct.

*6.1.1. Attacker.* Assume that the attacker $A$ has complete control over all communications during the run of $\Pi$. The capability of $A$ is essentially done by specifying the actions that he is allowed to perform, i.e., a group of the oracles he can query. Under the Bluetooth network setting, the interaction between the devices and $A$ is modeled by sending the queries to the oracles and receiving the results from the oracles. The oracles define how $A$ interacts with $\Pi$.

Launch $(i, j) \longrightarrow \{\text{sid}, \Pi_{i,j}, \Pi_{j,i}\}$: the Launch oracle means that the system initiates a unilateral authentication protocol run, where sid is set to a unique identifier for this run. $\Pi_{i,j}$ and $\Pi_{j,i}$ maintain the same $K_{\text{LINK}}$ and BD_ADDR. trans in both $\Pi_{i,j}$ and $\Pi_{j,i}$ are set to null, and $\delta$ in $\Pi_{i,j}$ is set to false.

Send $(m, \text{sid}, \Pi_{i,j}) \longrightarrow m'$ (resp. Send $(m, \text{sid}, \Pi_{j,i}) \longrightarrow m'$): the Send oracle sends a message $m$ to $i$ (resp., $j$) and receives the answer $m'$, which should be sent to the counterpart $j$ (resp., $i$). If $m$ is valid according to $\Pi_{i,j}$

(resp., $\Pi_{j,i}$), $m$ is added to $\Pi_{i,j}$'s (resp., $\Pi_{j,i}$'s) tran simultaneously. Herein, $m$, $m' \in \{1,0\}^* \cup \{\text{null}\}$, where null indicates that no message is transmitted.

Execute $(i, j) \longrightarrow \{\Pi_{i,j}, \Pi_{j,i}, \text{tran}, \text{sid}\}$: the Execute oracle is used to group one Launch query and successive use of the Send queries to execute a complete protocol run between $i$'s $\Pi_{i,j}$ and $j$'s $\Pi_{j,i}$. tran contains the transcript of all transmitted messages during this protocol run. Besides, the protocol run is identified by sid.

Result $(\Pi_{i,j}) \longrightarrow x$: the Result oracle can decide whether $\Pi_{i,j}$ successfully completes. That is, if $\Pi_{i,j}$'s $\delta$ is true, then $x = 1$; otherwise, $x = 0$.

Corrupt $(\Pi_{i,j}) \longrightarrow \{K_{\text{LINK}}\}$ (resp., Corrupt $(\Pi_{j,i}) \longrightarrow \{K_{\text{LINK}}\}$): the Corrupt oracle returns the link key $K_{\text{LINK}}$ secretly stored in $\Pi_{i,j}$ (resp., $\Pi_{j,i}$).

In the following, we present an experiment to define the protocol privacy by using the above oracles.

*6.1.2. Privacy.* As shown in Figure 6 we present the experiment $\text{Pri} - \text{Exp}_{\Pi,A}(k)$ to examine the privacy of the unilateral authentication protocol $\Pi$. In the setup stage, a set of devices are initiated by obtaining their $K_{\text{LINK}}$ and BD_ADDR. In the training I stage, the attacker $A$ can select any pair of the devices and learn the run of $\Pi$ by invoking the Launch, Send, Execute, Result, and Corrupt oracles. $A$ then chooses two uncorrupted devices $j_0$ and $j_1$ at his will and provides them to the Test oracle in the challenge stage. The Test oracle flips a coin bit $b \in \{0, 1\}$ and returns a device $j_b$ back to $A$. Then, to guess the $b$, $A$ can control the protocol runs between the claimant $j_b$ and any verifier $i$. That is, the training II stage continuously allows $A$ to access the Launch, Send, Execute, and Result oracles. Finally, $A$ should output his guessing of $b$. By this experiment, we propose the following definition for the protocol privacy.

*Definition 2* (privacy). A unilateral authentication protocol $\Pi$ is private if, for any probabilistic polynomial-time (PPT) attacker $A$, the guessing advantage,

$$\text{Pri} - \text{Adv}_{\Pi,A} = \left|\Pr\left[\text{Pri} - \text{Exp}_{\Pi,A}(k) = 1\right] - \Pr\left[\text{Pri} - \text{Exp}_{\Pi,A}(k) = 0\right]\right|, \tag{1}$$

is negligible in the security parameter $k$.

We use Definition 2 to examine the privacy of the unilateral authentication protocol as shown in Figure 2. The attacker $A$ can invoke the Execute oracle to record a tran = {AU_RAND, SRES} between a claimant $j$ and a verifier $i$ during the training I stage. Then, $A$ submits $j_0 = j$ and any other $j_1 \in I$ and calls the oracle Test $(j_0, j_1)$ in the challenge stage. During the training II stage, $A$ calls the oracle Launch $(i, j_b)$ to obtain $\Pi_{j_b,i}$ and sid and then invokes the oracle Send (AU_RAND, sid, $\Pi_{j_b,i}$) to receive the corresponding SRES. $A$ outputs the guess bit $b' = 0$ if the received SRES is equal to the recorded SRES during the training I stage. Otherwise, he outputs the guess bit $b' = 1$. Let $\nu$ be the probability that both the Send (AU_RAND, sid, $\Pi_{j_0,i}$) and the Send (AU_RAND, sid, $\Pi_{j_1,i}$) output the same SRES. We have

$$\begin{aligned}
\text{Pri} - \text{Adv}_{\prod,A} &= \left|\Pr\left[\text{Pri} - \text{Exp}_{\Pi,A}(k) = 1\right] - \Pr\left[\text{Pri} - \text{Exp}_{\Pi,A}(k) = 0\right]\right| \\
&= \Pr\left[\text{Pri} - \text{Exp}_{\prod,A}(k) = 1 | b = 0\right]\Pr[b = 0] + \Pr\left[\text{Pri} - \text{Exp}_{\prod,A}(k) = 1 | b = 1\right]\Pr[b = 1] \\
&\quad - \Pr\left[\text{PriExp}_{\prod,A}(k) = 0 | b = 0\right]\Pr[b = 0] - \Pr\left[\text{Pri} - \text{Exp}_{\prod,A}(k) = 0 | b = 1\right]\Pr[b = 1]| \\
&\leq \left|\frac{(1-\nu)}{2} + \frac{1}{2} - \frac{\nu}{2} - 0\right| = 1 - \nu.
\end{aligned} \tag{2}$$

Obviously, $\nu$ is negligible in the security parameter $k$. The protocol as shown in Figure 2 does not satisfy Definition 2 and therefore is not private. Consider the improved unilateral authentication protocol as shown in Figure 5. Although the attacker $A$ can intercept the previous values $\text{AU\_RAND}_V$, $\text{AU\_RAND}_C$, and SRES exchanged between Device 1 and Device 2, he cannot reuse them to identify the target Device 1. In the subsequent protocol run, $A$ can replay the previous $\text{AU\_RAND}_V$ to Device 1. However, Device 1 should compute a different SRES because a different $\text{AU\_RAND}_C$ is generated as an input of the algorithm $E_1$. As a result, $A$ cannot determine the identity of Device 1 by comparing the received SRES with the one recorded in the past protocol run. Certainly, the above privacy discussion is

**The experiment Pri – $\mathrm{Exp}_{\Pi,A}$ (k)**

**Stage 1. Setup**

(1) Run the pairing and link key generation procedure G $(1^k) \longrightarrow K_{\mathrm{LINK}}$.

(2) Write each link key $K_{\mathrm{LINK}}$ and corresponding Bluetooth-enabled device address BD_ADDR into each $i$, where $i \in I = \{1, 2, \ldots, n\}$.

**Stage 2. Training I**

The attacker A can adaptively issue the oracles $O = \{\mathrm{Launch}, \mathrm{Send}, \mathrm{Execute}, \mathrm{Result}, \mathrm{Corrupt}\}$ to any pair of the verifier $i$ and the claimant $j$, where $i, j \in I$.

**Stage 3. Challenge**

(1) The attacker $A$ selects two uncorrupted claimants $j_0$ and $j_1$, where $j_0, j_1 \in I$.

(2) The attacker $A$ calls an oracle Test $(j_0, j_1)$.

(3) To answer the query, the oracle Test flips a random coin bit $b$ and returns $j_b$ to the attacker $A$.

**Stage 4. Training II**

The attacker $A$ can adaptively issue the oracles $O = \{\mathrm{Launch}, \mathrm{Send}, \mathrm{Execute}, \mathrm{Result}\}$ to interact with the pair of any verifier $i$ and the claimant $j_b$, where $i \in I$ and the attacker $A$ is not allowed to invoke the Corrupt oracle for $\Pi_{i,j_b}$ or $\Pi_{j_b,i}$ in the training I stage.

**Stage 5. Guess**

The attacker $A$ outputs his guess bit $b'$ for the random bit $b$.

**Output.** If $b = b'$, then the experiment $\mathrm{Pri} - \mathrm{Exp}_{\Pi,A}(k)$ outputs 1; otherwise, the experiment $\mathrm{Pri} - \mathrm{Exp}_{\Pi,A}(k)$ outputs 0.

Figure 6: Privacy experiment for Bluetooth unilateral authentication protocol.

informal. In the following, we will present the privacy result of the improved protocol under our formal model.

*6.2. Privacy Property and Its Proof.* To evaluate the privacy of the improved unilateral authentication protocol, we need to use the keyed pseudorandom function assumption [38]. A keyed function $F$ receives for input some $K \in \{0, 1\}^k$ and $m \in \{0,1\}^*$ and outputs some $h \in \{0,1\}^*$. Here, $K$ is the key chosen uniformly at random. $F$ is a keyed pseudorandom function such that no polynomial-time distinguisher $D$ can detect if it is given a string sampled according to $F$ or a real random function $f$. The formal definition is given as follows.

*Definition 3.* Let $F$: $\{0, 1\}^k \times \{0, I\}^{l_1} \longrightarrow \{0, 1\}^{l_2}$ be an efficient keyed function. We say $F$ is a keyed pseudorandom function if, for all PPT distinguishers $D$, there exists a negligible function $\varepsilon$ such that

$$\left|\Pr\left[D\left(1^k, F_K(\,)\right) = 1\right] - \Pr\left[D\left(1^k, f(\,)\right) = 1\right]\right| \leq \varepsilon(k), \quad (3)$$

where the $k$-bit key $K$ is chosen uniformly at random and $f$ is chosen uniformly at random from the set of random functions mapping $l_1$-bit strings to $l_2$-bit strings.

Note that $D$ in Definition 3 has oracle access to the function in question (either $F$ or $f$). That is to say, $D$ is allowed to query the oracle at any time $x$, in response to which the oracle returns the value of the function evaluated at $x$. Finally, $D$ outputs 1 if it makes a correct guess. Now, we have the following theorem.

**Theorem 1.** *Let $\Pi$ be the improved unilateral authentication protocol as shown in Figure 5. If the algorithm $E_1$ in $\Pi$ is a keyed pseudorandom function and the k-bit link key $K_{LINK}$ is kept secret, then $\Pi$ is private in k under Definition 2.*

*Proof.* We know that the Corrupt oracle cannot help the attacker $A$ to guess the random bit $b$ in the experiment $\mathrm{Pri} - \mathrm{Exp}_{\Pi,A}(k)$. The reason is that all $K_{\mathrm{LINK}}$s of the pairing devices are independent and the oracle Corrupt is not allowed if the corresponding $K_{\mathrm{LINK}}$ is used in the training II stage. Hence, we do not consider the Corrupt oracle in the following discussions.

During the training II stage in the experiment $\mathrm{Pri} - \mathrm{Exp}_{\Pi,A}(k)$, the attacker $A$ can interact with the claimant $j_b$. We specify a simulator Sim to simulate $j_b$'s behavior in each run of $\Pi$ in this stage. However, Sim has no knowledge of the value of the random bit $b$ or the link key $K_{\mathrm{LINK}}$ in $\Pi$. We demonstrate that $A$'s interaction with Sim will be computationally indistinguishable from a real interaction with $j_b$. This means that $A$ cannot identify $j_b$ at the guess stage because $A$ gains no knowledge from its interaction with $j_b$ by the runs of $\Pi$.

Recall that the attacker $A$ selects $j_0$ and $j_1$ in the challenge stage of the experiment $\mathrm{Pri} - \mathrm{Exp}_{\Pi,A}(k)$. Let $L$ be the full list of the session transcript tran related to both $j_0$ and $j_1$ in the training I stage. Let $L'$ be the full list of the session transcript tran of $j_b$ in the training II stage. When $A$ invokes the Launch, Send, Execute, and Result oracles during the training II stage, Sim simulates the four oracles as follows.

Launch oracle: when the attacker $A$ calls Launch $(i, j_b)$, Sim generates its sid, $\Pi$ $(1^k, i, j_b, \mathrm{null})$, and $\Pi$ $(1^k, j_b, i, \mathrm{null})$ and then sends them to $A$. Here, $\Pi$ $(1^k, i, j_b, \mathrm{null})$ and $\Pi$ $(1^k, j_b, i, \mathrm{null})$ are, respectively, used to simulate $\Pi_{i,j_b}$ and $\Pi_{j_b,i}$ and null means that Sim does not know the link key $K_{\mathrm{LINK}}$.

Send oracle: (1) When the attacker $A$ calls Send (null, sid, $\Pi$ $(1^k, i, j_b, \mathrm{null})$), Sim randomly generates the $\mathrm{AU\_RAND}_V$ as the verifier $i$ in $\Pi$, records the $\mathrm{AU\_RAND}_V$ in its $L'$, and sends it to $A$. (2) When $A$ calls Send ($\mathrm{AU\_RAND}_V$, sid, $\Pi$ $(1^k, j_b, i, \mathrm{null})$) and $\mathrm{AU\_RAND}_V$ is generated by Sim, Sim generates the random $\mathrm{AU\_RAND}_C$ and the random SRES itself and records them in its $L'$ and then sends the $\mathrm{AU\_RAND}_C$ and SRES to $A$. Sim terminates the protocol run if $A$ calls Send ($\mathrm{AU\_RAND}_V$, sid, $\Pi$ $(1^k, j_b, i, \mathrm{null})$) and Sim does not generate $\mathrm{AU\_RAND}_V$. (3) When $A$ calls Send ({$\mathrm{AU\_RAND}_C$, SRES}, sid, $\Pi$ $(1^k, i, j_b, \mathrm{null},$

AU_RAND$_V$)), Sim sends the "accept" decision to $A$ if {AU_RAND$_V$, AU_RAND$_C$, SRES} has existed in its $L'$; otherwise, Sim sends the "reject" decision to $A$.

Execute oracle: to simulate Execute ($i$, $j_b$), Sim generates sid, AU_RAND$_V$, AU_RAND$_C$, and SRES just like the Launch oracle and the Send oracle; records {AU_RAND$_V$, AU_RAND$_C$, SRES} in its $L'$; and then sends them to $A$.

Result oracle: when the attacker $A$ invokes Result ($\Pi$ ($1^k$, $i$, $j_b$, null, tran)), Sim returns 1 to $A$ if the tran is in its $L'$; otherwise, it returns 0 to $A$. In the case of calling Result ($\Pi$ ($1^k$, $i$, $j_b$, $K_{\text{LINK}}$, tran)), Sim should replay the query to $i$ or $j_b$ and return the corresponding response to $A$.

To distinguish Sim's training II stage from a real training II stage, the attacker $A$ must be able to identify at least one invalid session between the claimant $j_b$ and any other verifier $i$. In other words, $A$ must rule out at least one tran = {AU_RAND$_V$, AU_RAND$_C$, SRES} in order to determine that Sim is present during the training II stage. Assume that $A$, respectively, makes at most $q$ ($k$) queries to the Send oracle and the Execute oracle in each training stage of the experiment $\text{Pri} - \text{Exp}_{\Pi,A}$ ($k$), where $q$ ($k$) is a polynomial function. Consequently, one of the following two cases must occur at some point during the experiment $\text{Pri} - \text{Exp}_{\Pi,A}$ ($k$).

Case 1: there are two session transcripts: $\{\text{AU.RAND}_V^L, \text{AU.RAND}_C^L, \text{SRES}^L\} \in L$ and $\{\text{AU.RAND}_V^{L'}, \text{AU.RAND}_C^{L'}, \text{SRES}^{L'}\} \in L'$ such that $\text{AU.RAND}_V^L = \text{AU. RAND}_V^{L'}$ and $\text{AU.RAND}_C^L = \text{AU.RAND}_C^{L'}$. We know that $\text{SRES}^{L'}$ is randomly generated by Sim. Hence, the attacker $A$ can figure out Sim by verifying whether $\text{SRES}^L$ is equal to $\text{SRES}^{L'}$. Let $|L|$ and $|L'|$ denote, respectively, the number of the session transcripts in $L$ and $L'$. We have $|L| \le q$ ($k$) and $|L'| \le q$ ($k$) because $A$ makes, respectively, at most $q$ ($k$) Send and Execute calls in each corresponding training stage. $A$ can control the $\text{AU.RAND}_V^L$ and $\text{AU.RAND}_V^{L'}$; however, $\text{AU.RAND}_C^L$ and $\text{AU.RAND}_C^{L'}$ are, respectively, generated by the claimant $j_b$ and Sim. Since the values $\text{AU.RAND}_C^L$ and $\text{AU.RAND}_C^{L'}$ are all random $k$-bit values, Case 1 occurs with the probability at most $q$ $(k)^2/2^k$.

Case 2: Sim randomly chooses SRES in the session. Comparatively, the $j_b$ should use the algorithm $E_1$ to calculate SRES. Hence, the attacker $A$ depends on whether SERS is randomly chosen or calculated to recognize Sim. We know that the algorithm $E_1$ is a keyed pseudorandom function (see Definition 3). According to (3), the probability $u$ that $A$ can pick out Sim is

$$u(k) \le \left|\Pr\left[D\left(1^k, F_K(\,)\right) = 1\right] - \Pr\left[D\left(1^k, f(\,)\right) = 1\right]\right| \le \varepsilon(k). \tag{4}$$

Hence, the polynomially bounded attacker $A$ can distinguish Sim from the real $j_b$ with the negligible probability at most $q$ $(k)^2/2^k + u$ $(k) \le q$ $(k)^2/2^k + \varepsilon$ $(k)$.

Due to Theorem 1, the improved unilateral authentication protocol can prevent the attacker from identifying, tracing, or linking to the device, if and only if the device does not compromise its secret parameter $K_{\text{LINK}}$ stored in the memory. □

## 7. Other Countermeasures against Proposed Weakness

*7.1. Limiting Authentication Failure Attempts.* Clearly, the proposed privacy weakness can be partly repaired if the claimant limits the number of failed authentication attempts for each link key. We know the claimant can confirm the authentication failure because the verifier should terminate its subsequent communication with the claimant. Hence, a countermeasure is to change its link key whenever FA > $t$, where FA is the number of failed authentication attempts since the link key was last changed and $t$ is the threshold number of failed attempts. If $t = 0$, then the attacker can at most track the identity of the claimant one time. However, if the value of $t$ is too small, it may lead to the high overheads of the pairing devices. The reason is that the device's hardware and software faults may also cause authentication failure attempts. In this situation, the pairing devices need frequently to update their link key by running the pairing and link key generation procedure.

*7.2. Avoiding Unilateral Authentication Protocol.* To overcome the proposed privacy weakness, it is required that both pairing devices always execute the mutual authentication protocol instead of the unilateral authentication protocol. However, this countermeasure has disadvantages as follows.

The mutual authentication protocol is performed, only if both paring devices support the mutual authentication protocol. Otherwise, the unilateral authentication protocol is performed. In addition, during the link initialization stage, the attacker can also cheat the paring devices to be backward compatible with the unilateral authentication protocol. One way of deceiving is by intercepting and modifying the Bluetooth data packets of the device capabilities.

The mutual authentication protocol is merely used when the link key has been generated using SSP with the $P-256$ Elliptic Curve. Comparatively, the unilateral authentication protocol supports SSP with not only the $P-256$ Elliptic Curve but also the $P-128$ Elliptic Curve.

The unilateral authentication suffices for many Bluetooth MCS cases. Further, it is more desirable in the Bluetooth MCS, since the mutual authentication incurs more implementation costs compared with the unilateral authentication.

## 8. Conclusions

We investigate the traceability weakness relating to the unilateral authentication protocol in the Bluetooth standard, which is widely applied in the Bluetooth-enabled devices. The authentication service should provide strong privacy protection in fear of vulnerabilities and abuse mounting. Such privacy requirement for the weak authentication protocol is reinforced by the possibility of more and more Bluetooth-enabled device abuses in the real world, such as the Bluetooth MCS. An improved unilateral authentication protocol is thus proposed to overcome the weakness of traceability. Our improved protocol is simple and easy to implement in Bluetooth-enabled devices, though with a little extra overheads. In addition, non-protocol countermeasures are also suggested to fix the weakness of traceability. Certainly, the unilateral authentication protocol is still unable to assure the overall privacy of the Bluetooth-enabled devices and their networks. Nevertheless, we believe that our results are a steady step toward enhancing Bluetooth security solutions. Our future work is to study the privacy problems of the mutual authentication protocol and the confidentiality procedure in accordance with the Bluetooth standard.

## Data Availability

The data used to support the findings of this study are included in the article.

## Conflicts of Interest

The authors declare that they have no known conflicts of interest or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

The work of Dr. Da-Zhi Sun was funded in part by the National Natural Science Foundation of China under Grant no. 61872264.

## References

[1] S. Zeadally, F. Siddiqui, and Z. Baig, "25 years of Bluetooth technology," *Future Internet*, vol. 11, no. 9, Article ID 194, 2019.

[2] S. S. Hassan, S. D. Bibon, M. S. Hossain, and M. Atiquzzaman, "Security threats in Bluetooth technology," *Computers & Security*, vol. 74, pp. 308–322, 2018.

[3] Bluetooth SIG Proprietary, *Specification of the Bluetooth System, Covered Core Package Version: 5.1, Master Table of Contents & Compliance Requirements*, Bluetooth SIG Proprietary, Kirkland, WA, USA, 2019, https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=457080.

[4] Bluetooth SIG Proprietary, *Specification of the Bluetooth System, Core Specification Version: 5.2, Master Table of Contents & Compliance Requirements*, Bluetooth SIG Proprietary, Kirkland, WA, USA, 2019, https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=478726.

[5] Bluetooth SIG Proprietary, *Core Specification Supplement Version: 9, Master Table of Contents & Compliance Requirements*, Bluetooth SIG Proprietary, Kirkland, WA, USA, 2019, https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=480305.

[6] J. K. Jung and T. M. Chung, "Enhancing PIN input for preventing eavesdropping in BLE legacy pairing," in *Proceedings of International Conference on Computer Science and Its Applications-CSA'15*, pp. 813–817, Cebu, Philippines, January 2015.

[7] D. Z. Sun and X. H. Li, "Vulnerability and enhancement on Bluetooth pairing and link key generation scheme for security modes 2 and 3," in *Proceedings of the 18th International Conference on Information and Communications Security-ICICS'16*, pp. 403–417, Singapore, November 2016.

[8] K. Haataja and P. Toivanen, "Two practical man-in-the-middle attacks on Bluetooth secure simple pairing and countermeasures," *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 384–392, 2010.

[9] D. Z. Sun, Y. Mu, and W. Susilo, "Man-in-the-middle attacks on secure simple pairing in Bluetooth standard v5.0 and its countermeasure," *Personal and Ubiquitous Computing*, vol. 22, no. 1, pp. 55–67, 2018.

[10] S. Gajbhiye, S. Karmakar, M. Sharma, and S. Sharma, "Bluetooth secure simple pairing with enhanced security level," *Journal of Information Security and Applications*, vol. 44, pp. 170–183, 2019.

[11] D. Z. Sun and L. Sun, "On secure simple pairing in Bluetooth standard v5.0—Part I: authenticated link key security and its home automation and entertainment applications," *Sensors*, vol. 19, no. 5, Article ID 1158, 2019.

[12] IEEE, 2002, IEEE standard for telecommunications and information exchange between systems-LAN/MAN-specific requirements—Part 15: wireless medium access control (MAC) and physical layer (PHY) specifications for wireless personal area networks (WPANs), IEEE std 802.15.1, https://ieeexplore.ieee.org/document/1016473.

[13] A. Basalamah, "Oppotunistic crowdsensing framework for Internet of things using Bluetooth low energy," *Journal of Engineering*, vol. 2016, no. 5, pp. 141–146, 2016.

[14] A. Draghici and M. Van Steen, "A survey of techniques for automatically sensing the behavior of a crowd," *ACM Computing Surveys*, vol. 51, no. 1, Article ID 21, 2018.

[15] J. Padgette, J. Bahr, M. Batra et al., *Guide to Bluetooth Security*, National Institute of Standards and Technology, Gaithersburg, MD, USA, 2017, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-121r2.pdf.

[16] F. L. Wong and F. Stajano, "Location privacy in Bluetooth," in *Proceedings of the 2005 European Workshop on Security and Privacy in Ad Hoc and Sensor Networks-ESAS'05*, pp. 176–188, Visegrad, Hungary, January 2005.

[17] H. Kikuchi and T. Yokomizo, "Location privacy vulnerable from Bluetooth devices," in *Proceedings of the 16th International Conference on Network-Based Information Systems-NBiS'13*, pp. 534–538, Gwangju, Republic of Korea, September 2013.

[18] M. Cominelli, F. Gringoli, P. Patras, M. Lind, and G. Noubir, "Even black cats cannot stay hidden in the dark: full-band de-anonymization of Bluetooth classic devices," in *Proceedings of the 2020 IEEE Symposium on Security and Privacy-SP'20*, pp. 534–548, San Francisco, CA, USA, May 2020.

[19] J. Gibbs, "BLE and Laird's BL6x0 series & BT900 modules: a guide to security and privacy," A White Paper from Laird, 2014.

[20] Cypress, *AN99209, PSoC®4 BLE and PRoC—BLE: Bluetooth LE 4.2 Features*, Cypress, San Jose, CA, USA, 2021, https://www.cypress.com/file/224826/download.

[21] P. Wang, "Bluetooth low energy-privacy enhancement for advertisement," Ph.D. dissertation, Norwegian University of Science and Technology, Trondheim, Norway, 2014.

[22] A. K. Das, P. H. Pathak, C. N. Chuah, and P. Mohapatra, "Uncovering privacy leakage in BLE network traffic of wearable fitness trackers," in *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications-HotMobile'16*, pp. 99–104, St Augustine, FL, USA, Feburary 2016.

[23] T. Issoufaly and P. U. Tournoux, "BLEB: Bluetooth low energy botnet for large scale individual tracking," in *Proceedings of the 1st International Conference on Next Generation Computing Applications-NextComp'17*, pp. 115–120, Mauritius, August 2017.

[24] A. Korolova and V. Sharma, "Cross-app tracking via nearby Bluetooth low energy devices," in *Proceedings of the 8th ACM Conference on Data and Application Security and Privacy-CODASPY'18*, pp. 43–52, Tempe, Arizona, USA, March 2018.

[25] J. K. Becker, D. Li, and D. Starobinski, "Tracking anonymized Bluetooth devices," *in Proceedings on Privacy Enhancing Technologies*, vol. 2019, no. 3, pp. 50–65, 2019.

[26] G. Celosia and M. Cunche, "Detecting smartphone state changes through a Bluetooth based timing attack," in *Proceedings of the 11th ACM Conference on Security and Privacy in Wireless and Mobile Networks-WiSec'18*, pp. 154–159, Stockholm, Sweden, June 2018.

[27] W. Albazrqaoe, J. Huang, and G. Xing, "A practical Bluetooth traffic sniffing system: design, implementation, and countermeasure," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 71–84, 2019.

[28] E. Bello-Ogunu, M. Shehab, and N. S. Miazi, "Privacy is the best policy: a framework for BLE beacon privacy management," in *Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference-COMPSAC'19*, pp. 823–832, Milwaukee, Wisconsin, USA, July 2019.

[29] D. Z. Sun, L. Sun, and Y. Yang, "On secure simple pairing in Bluetooth standard v5.0—Part II: privacy analysis and enhancement for low energy," *Sensors*, vol. 19, no. 15, Article ID 3259, 2019.

[30] S. C. Cha, M. S. Chuang, K. H. Yeh, Z. J. Huang, and C. Su, "A user-friendly privacy framework for users to achieve consents with nearby BLE devices," *IEEE Access*, vol. 6, pp. 20779–20787, 2018.

[31] T. Wu, F. Wu, C. Qiu, J. M. Redouté, and M. R. Yuce, "A rigid-flex wearable health monitoring sensor patch for IoT-connected healthcare applications," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6932–6945, 2020.

[32] M. Whaiduzzaman, M. R. Hossain, A. R. Shovon et al., "A privacy-preserving mobile and fog computing framework to trace and prevent COVID-19 community transmission," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 12, pp. 3564–3575, 2020.

[33] Specification of the Bluetooth System, *Covered Core Package Version: 4.2, Master Table of Contents & Compliance Requirements*, Bluetooth SIG Proprietary, 2014, https://www.bluetooth.org/docman/handlers/DownloadDoc.ashx?doc_id=441541.

[34] P. P. Ray and S. Agarwal, "Bluetooth 5 and internet of things: potential and architecture," in *Proceedings of International Conference on Signal Processing, Communication, Power and Embedded System-SCOPES'16*, pp. 1461–1465, Paralakhemundi, India, October 2016.

[35] R. Y. Y. Chan, E. Sato-Shimokawara, X. Bai, M. Yukiharu, S. W. Kuo, and A. Chung, "A context-aware augmentative and alternative communication system for school children with intellectual disabilities," *IEEE Systems Journal*, vol. 14, no. 1, pp. 208–219, 2020.

[36] A. H. M. Aman, W. H. Hassan, S. Sameen, Z. S. Attarbashi, M. Alizadeh, and L. A. Latiff, "IoMT amid COVID-19 pandemic: application, architecture, technology, and security," *Journal of Network and Computer Applications*, vol. 174, Article ID 102886, 2021.

[37] A. Juels and S. A. Weis, "Defining strong privacy for RFID," *ACM Transactions on Information and System Security*, vol. 13, no. 1, Article ID 7, 2009.

[38] J. Katz and Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*, Chapman & Hall/CRC, Boca Raton, FL, USA, 2nd edition, 2014.