

Research Article

Data-Sharing Method for Multi-Smart Devices at Close Range

Myoungbeom Chung¹ and Ilju Ko²

¹*Division of Computer Engineering, Sungkyul University, Gyeonggi-do 430-742, Republic of Korea*

²*Department of Global Media, Soongsil University, Seoul 156-743, Republic of Korea*

Correspondence should be addressed to Myoungbeom Chung; nzin@sungkyul.ac.kr and Ilju Ko; andy@ssu.ac.kr

Received 8 July 2015; Revised 22 October 2015; Accepted 9 November 2015

Academic Editor: Kamal Deep Singh

Copyright © 2015 M. Chung and I. Ko. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We proposed a useful data-sharing method among multi-smart devices at close range using inaudible frequencies and Wi-Fi. The existing near data-sharing methods mostly use Bluetooth technology, but these methods have the problem of being unable to be operated using different operating systems. To correct this flaw, the proposed method that uses inaudible frequencies through the inner speaker and microphone of smart device can solve the problems of the existing methods. Using the proposed method, the sending device generates trigger signals composed of inaudible sound. Moreover, smart devices that receive the signals obtain the shared data from the sending device through Wi-Fi. To evaluate the efficacy of the proposed method, we developed a near data-sharing application based on the trigger signals and conducted a performance evaluation experiment. The success rate of the proposed method was 98.8%. Furthermore, we tested the user usability of the Bump application and the proposed method and found that the proposed method is more useful than Bump. Therefore, the proposed method is an effective approach for sharing data practically among multi-smart devices at close range.

1. Introduction

With the development of the mobile phone and communication technology, the existing mobile phone has changed into a smart device that has various functions, such as camera, mp3 player, and on/offline mobile games. Smart device users can share their daily lives and thoughts online through social network services (SNS) such as Facebook and Twitter. Users can share not only photos, such as those of their daily lives or vacation, but also data such as name card and work materials, among others, during business meetings. The data-sharing methods among smart devices are divided into those that can be used at close range using Bluetooth or Wi-Fi Direct and at a long distance through Wi-Fi and a server computer.

The data-sharing methods for close range use Bluetooth, Wi-Fi Direct, Airdrop, or NFC. Data can be sent directly without the use of extra service applications. The method that uses Bluetooth requires the same application to be installed on each smart device, and data sharing can occur through the pairing of smart devices [1]. The maximum speed of data transmission using Bluetooth is 24 Mbps, and it requires only the inner Bluetooth module of a smart device. However, as this method can only be used with the same operating system

(OS), it cannot transmit data between a smart device that uses an Android OS and a smart device that uses iOS. Wi-Fi Direct is the upgraded model of the Bluetooth feature for near data transmission and has a data transmission speed of 300 Mbps [2]. Therefore, it is more suitable for sending big data to multimedia devices such as cameras or printers than using data transmission technology among smart devices. Similarly, Apple Inc. created the Airdrop technology that uses Bluetooth instead of Wi-Fi Direct [3]. Airdrop can transmit data among smart devices at close range. However, similar to Bluetooth, this technology can only be used with the same OS. In addition, Android Beam and S Beam are newly released data sharing methods. They can work well with smart devices with built-in NFC. Android Beam can share much data, such as address lists, URLs of web browsers, GPS data of Google Maps, market addresses of the Play store, and applications downloaded from the Android market [4, 5]. S Beam can share media data, such as mp3 music files, photo images, and movie files [6, 7]. However, Android Beam and S Beam cannot work with smart devices without NFC, such as iPhones and iPads.

Data-sharing methods for long distance use Wi-Fi and a server computer. Users are required to use extra services

or SNS and become friends with other users to be able to share data. The typical applications for long-distance data sharing are Kakao Talk, Whatsapp, and Viber, among others [8–10]. These applications can be used for sharing data and exchanging photo files between two friends. The advantage of this method is that smart device users can share data at a long distance, such as when users are in different countries. However, these data-sharing methods for long distance have the disadvantage of being unable to send data to any smart device if the user does not join the application. Users must be friends with each other to share data as a pretask. To overcome these disadvantages, Bump Technologies created the Bump application in 2010 [11]. The Bump application uses Wi-Fi, global positioning system (GPS) information, and acceleration sensor to share data at close range. This technology can share data among smart devices without joining any application or becoming friends with other users. Moreover, it can be used with different OS. As Bump Technologies use Open API and the Bump server, the Bump application can be used for the various applications of smart phones. However, Bump can only conduct 1:1 data sharing. If three smart devices are used to share data in the same place and at the same time, the Bump application will not work as the smart device that sends the data cannot know which of the other smart devices will receive the data.

Therefore, we propose a useful data-sharing method among multi-smart devices at close range that can solve the problems of the existing methods. The proposed method uses inaudible frequencies as a trigger signal, instead of the shaking motion of the Bump application, and Wi-Fi and GPS information for sharing data among smart devices at close range. The audible range known scientifically is 20 Hz to 22,000 Hz. However, the audible range for most people is below 20,000 Hz [12, 13], and the older the people are, the lower their audible range is. People in the 40s and 50s can hear frequencies below 13,000 Hz [14]. Therefore, the proposed method uses a frequency of over 18,000 Hz. The smart device (sending device) that shares data generates a trigger signal from an inner speaker using a combination of inaudible frequencies. At the same time, the sending device sends the shared data and its current GPS information to the sharing server. Smart devices (receiving devices) are located around the sending device and receive the data-sharing process surrounded by sound through an inner microphone. If the receiving device perceives the trigger signal assigned as the data-sharing signal, it sends its current GPS information to the server and downloads the shared data from the server. We upgrade the high-frequency signal as the trigger signal rather than the existing high-frequency signal in the literature. As the trigger signal of the proposed method uses two or more high frequencies, we can upgrade the reaching distance of the trigger and the accuracy of data transmission.

We developed a data-sharing application based on a smart device and conducted a data-sharing experiment according to distance to evaluate whether the proposed method is suitable for data sharing among smart devices at close range. The result of the experiment showed 98% accuracy in

the data-sharing test within a 5 m distance irrespective of the OS of the smart devices. We then conducted a 1:1 and 1: N sharing accuracy test as a comparative experiment using the Bump application. We surveyed the experiment participants to evaluate the usability of the proposed method. As the Bump application has not supported the Bump API and Bump server since January 31, 2014, we developed an application that works like the Bump application based on the Bump algorithm, and we set up a server like the Bump server to share data. The results of the sharing accuracy experiment and of the survey on usability showed that the proposed method was better than the Bump application. Therefore, the proposed method could be a useful method for sharing data effectively among multi-smart devices at close range.

The present paper is organized as follows. In Section 2, we explain the Bump application that is useful in sharing data at close range among smart devices and the existing research that uses high frequency, which we use as the trigger signal in the proposed method. In Section 3, we present the general architecture of the proposed method for sharing data among multi-smart devices, the process of the trigger signal using high frequency, and the sharing server. In Section 4, we show the data-sharing application that applies the proposed method and report the experimental results of the data transmission accuracy and usability of the application. We conclude the study in Section 5.

2. Previous Work

This section briefly explains the existing methods for near data-sharing technology and presents the general architecture of the Bump application. We explain the existing research that uses high frequency to send information. Although near data-sharing technologies among smart devices use Bluetooth, Wi-Fi Direct, Airdrop, and NFC, among others, these technologies can only use the same OS [15–17]. To solve this problem, Bump Technologies created the Bump application and opened the Bump API. This technology has been adopted in various smart device applications of iOS and Android since 2012 [18–20].

The Bump application uses three inner sensors of the smart device, unlike Bluetooth, Wi-Fi Direct, and Airdrop. Three inner sensors are the accelerometer, GPS, and Wi-Fi. The accelerometer detects the user's shaking action to begin data sharing, and the GPS obtains the location information of each device [21, 22]. The sending device sends the location information and shared data to the Bump server through Wi-Fi, while the receiving device downloads the shared data from the Bump server also through Wi-Fi. Therefore, the Bump application involves two smart devices and one Bump server. The general architecture of the Bump application is shown in Figure 1.

In Figure 1, we redraw the workflow of the Bump API, which Ahren suggested [23]. If the sending device and the receiving device do the shaking action to share data, each smart device sends its GPS information through 3G, long-term evolution (LTE), or Wi-Fi to the Bump server (©). The Bump server assesses whether the sending device and the

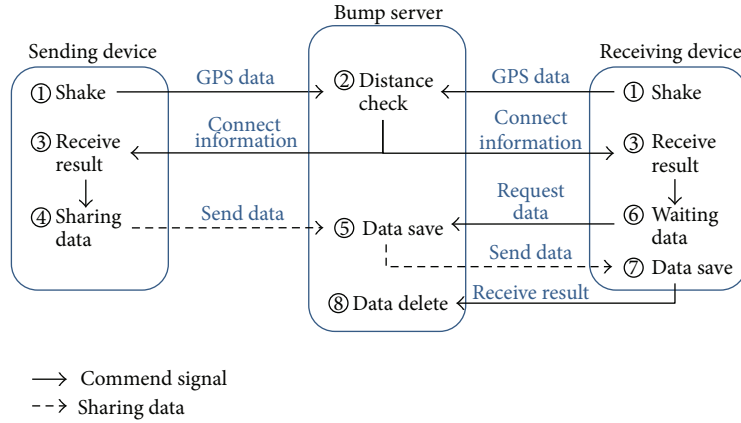


FIGURE 1: General architecture of the Bump application.

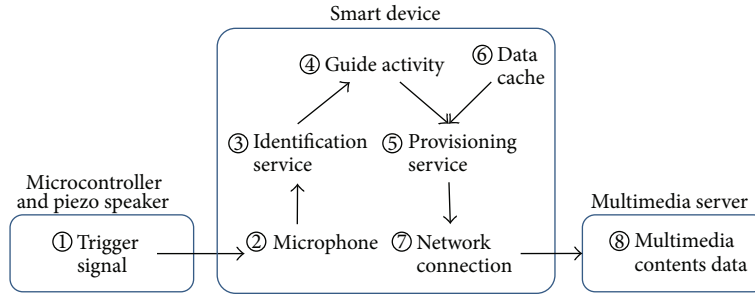


FIGURE 2: General architecture of the SmartGuide.

receiving device are close using the GPS information of each smart device and connecting time to the Bump server (2). The Bump server then sends the connect information to each smart device on whether the shaking action is successful (3). If the connect information is successful, the sending device sends the shared data to the Bump server (4), which saves the data (5). At this time, the receiving device requests the shared data and waits to save it (6). When the Bump server sends the data, the receiving device begins to save the data in its memory space through 3G, LTE, or Wi-Fi (7). Finally, when the transmission of the shared data ends, the receiving device sends the transmission result of the shared data to the Bump server, which deletes the shared data from the server (8).

The Bump application can share data easily among smart devices at close range regardless of the OS of the smart devices. It began to support the data-sharing service between a computer and a smart device in 2013 [24]. The sharable data formats of the Bump application are picture, video, and contact information, among others. By 2013, it had been downloaded 125 million times [25]. However, the Bump application requires the coinciding shaking action of the sending device and the receiving device, and it can only be used for 1:1 and not for 1:N devices.

The existing methods that use high frequency are indoor positioning technology and indoor information transmission technology. Indoor positioning technology is a user position trace technology that uses a smart device inside a room

or a building where the GPS of a smart device does not work well. Viacheslav proposed an indoor position trace algorithm based on high frequency using one mobile phone and four microphones [12]. He tested the output performance of the speaker of each mobile phone with various smart devices to determine the inaudible high frequency, which is suitable for a user's position trace. Bihler et al. suggested a trigger signal using the high frequency and developed a SmartGuide that could support user-related information according to a user's indoor position. Figure 2 shows that we redrew the general architecture of Bihler et al.'s SmartGuide [26].

In Figure 2, the trigger signal is generated by an 8-bit 3.2 MHz Freescale microcontroller and a simple piezo speaker (1). Bihler et al. used 20 kHz and 22 kHz as the high frequencies for the trigger signal, which could send 8-bit data at 208 ms. The smart device confirms the identification service through a microphone (2, 3) and runs the guide activity (4). If the smart device already has the information in the data cache (6), it sets the provisioning service (5). However, if the smart device does not have the information, the smart device connects to the multimedia server through 3G, LTE, or Wi-Fi to download the multimedia content (7, 8) and sets the provisioning service. Bihler's method is suitable for quiet indoor spaces or museum where GPS does not work well. However, as the trigger signal of this method requires many changes in high frequency for a short time, the trigger signal generates some

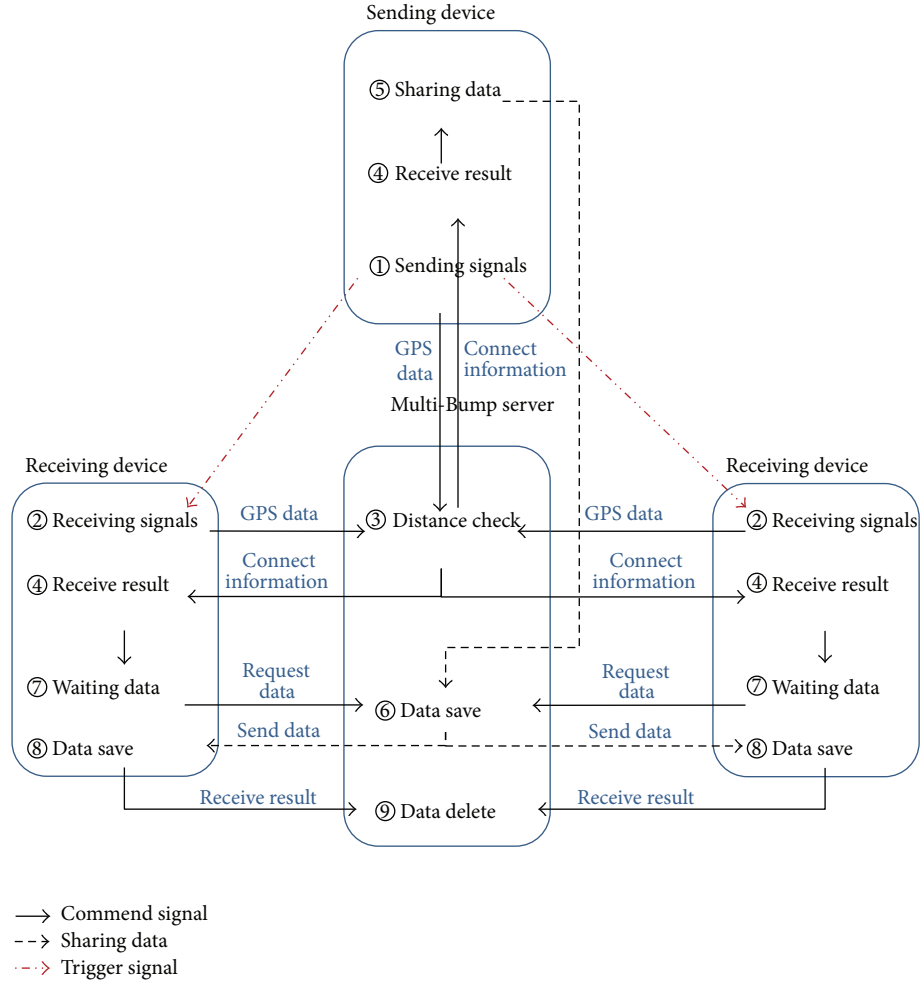


FIGURE 3: General architecture of the data-sharing method using high frequency as the trigger signal.

noise and the transmission accuracy of the trigger signal is low.

Next, the existing methods that use high frequency as little information transmission technology indoors were SonicData proposed by Nittala and Authentication method proposed by Kim [27, 28]. However, those methods have a problem that they need so many times to transmit a small quantity of data.

3. Data-Sharing Method for Multi-Smart Devices

This section explains the general architecture of the proposed data-sharing method among multi-smart devices using inaudible high frequency. The processing method of the trigger signals using high frequency and the sharing server is also presented. Figure 3 shows the general architecture of the proposed method. As shown in Figure 3, the sending device generates a trigger signal to share data and sends its own GPS information to the Multi-Bump server (①) at the same time. The nearby receiving devices analyze the surrounding sounds using the inner microphone in the smart device. When the

receiving devices detect the trigger signal, the devices send their own SPG information to the Multi-Bump server (②). Then, the Multi-Bump server analyzes the received GPS data from each device and calculates the distance among the smart devices (③).

The Multi-Bump server checks the distance interval and sends the connect information according to the distance check to the sending device and the receiving devices (④). When the sending device receives the result that the data have been shared to all available devices, the sending device uploads the shared data to the Multi-Bump server (⑤). The Multi-Bump server then saves the uploaded shared data temporarily (⑥). When the waiting receiving devices request the shared data (⑦), the Multi-Bump server sends the shared data to each receiving device. The receiving devices begin to save the shared data in their respective memories (⑧), and each device sends the result of the received data to the Multi-Bump server when saving of the shared data ends. Finally, when the Multi-Bump server checks all the received results from the receiving devices, it deletes the shared data (⑨).

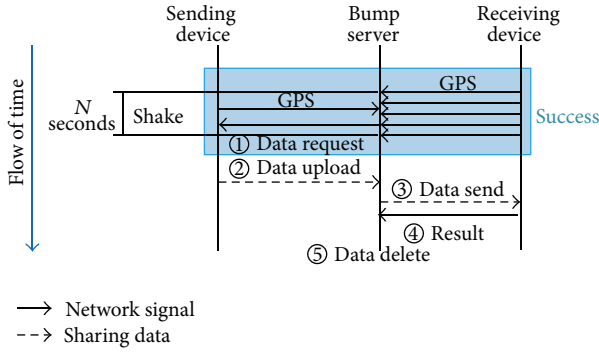


FIGURE 4: General data-sharing flow of the Bump application.

The difference between the proposed method and the existing Bump method is denoted by ① and ② movements in Figure 3. The Bump method uses a shaking action as the data-sharing signal. Moreover, it can share data only when each smart device shakes at the same time, as shown in Figure 4. Conversely, as the proposed method uses inaudible frequency as the data-sharing signal instead of a shaking action, it can share data with several smart devices at the same time. As shown in Figure 4, the Bump method can share data when the sending device and the receiving device shake within N seconds. In other words, if shake permission time of the Bump API is N seconds, the receiving device must shake within $\pm N/2$ seconds from the shaking time T of the sending device to share data.

Then, the sending device and the receiving device progress from ① to ⑤ in order and they share the data. However, if the shaking time is over, as shown in Figure 5, the Bump method for sharing data fails. In Figure 5, the reason for the first failure is that the receiving device performs the shaking action too early.

As the receiving device does the shaking action before $T - N/2$ seconds, the Bump method fails (①). The reason for the second fail is that the receiving device does the shaking action too late. As the receiving device does the shaking action after $T + N/2$ seconds, the Bump method fails (②). Conversely, the proposed method can solve this problem because the sending device generates the trigger signal using inaudible frequency for sharing data consistently until the receiving device detects the signal. Figure 6 shows the flow of the proposed method using inaudible frequency for data sharing.

As shown in Figure 6, we do not use N seconds, which is used in the Bump method. The sending device generates the trigger signal consistently instead of N seconds until the receiving device responds to detecting the trigger signal and sends its GPS information to the Multi-Bump server. When the receiving device detects the trigger signal, it sends its own GPS information to the Multi-Bump server. At this time, the server calculates the distance from the GPS information of each device and requests the shared data for the sending device according to the connection result (①). The sending device that requests the shared data stops to generate the trigger signal and continues the

data sharing from step ③ to step ⑤. Furthermore, the proposed method can share data from one sending device to many receiving devices at the same time. Figure 7 shows the data-sharing flow of one sending device and two receiving devices.

In Figure 7, the sending device generates the trigger signal (①). At the same time, it sends its own GPS information to the Multi-Bump server and is waiting for the request for sharing data from the server (②). The receiving devices that detect the trigger signal send the GPS information of each device to the Multi-Bump server (③), which calculates the distance from the GPS information of all devices. If the connection result is successful, the server requests the sending of shared data from the sending device (④). Then, the server saves the number of receiving devices temporarily and sends the shared data, which are received by each receiving device from the sending device (⑤, ⑥). When the receiving devices obtain the shared data, the receiving devices send a message about the transmission result of the shared data to the server (⑦). The server counts the number of messages about the transmission result. If the number of the receiving devices is equal to the number of messages, the server deletes the shared data uploaded from the sending device (⑧). Therefore, the proposed method can share data easily in 1 : N because it does not need to consider the time of the shaking action.

Next, the trigger signal used in the proposed method is applied to the control signal proposed by Chung and Choo [29]. Chung and Choo used two base signals and one low-latency key as the trigger signal. Two base signals are fixed high frequencies and the low-latency key is a high frequency that can be changed by a mobile central processing unit (CPU). Chung's method is used to determine why the arrival distance of the trigger signal is so far and why its accuracy is high. Therefore, we use two base signals and one low-latency key and the frequency as the trigger signal ranges from 18 kHz to 22 kHz. However, as the length of the trigger signal used in Chung and Choo's study is short, we make the trigger signal longer and generate it many times to send the trigger signal to various smart devices. To avoid distinguishing the trigger signal from the surrounding noise, we use the changeable low latency as the sharing key value in near data sharing. The pseudocode in Pseudocode 1 is the detecting code of the trigger signal applied to the sharing key using the low-latency. It also works in the receiving smart devices. In Pseudocode 1, k is the parameter for counting the trigger signal and A_t is the audio data obtained by the inner microphone of the smart device during t time. A_t is separated from each frequency F_t as fast Fourier transform (FFT). If the base signals and the low-latency key of the F_t values exist, k increases. If the base signals and the low-latency key of the F_t values do not exist, k sets the initialization as zero.

When the k parameter is over the threshold value α , the receiving device makes an assessment whether to receive the trigger signal and sends its own GPS information to the Multi-Bump server. Then, the receiving device requests the shared data and ends the process to detect the trigger signal.

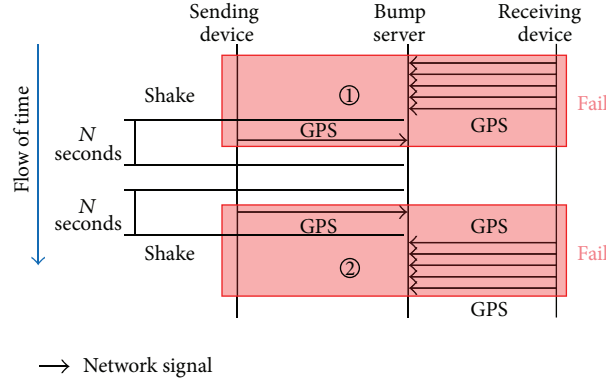


FIGURE 5: Example of a data sharing failure using the Bump method.

```

k ← 0;
While (receive  $A_t$ ) Do
   $F_t \leftarrow \text{FFT}(A_t)$ ;
  If ( $F_t$  has the base signals) Then
    If ( $F_t$  has key low-latency) Then
      k ← k + 1;
      If (k is more than threshold  $\alpha$ ) Then
        GPS data send to server;
        Request sharing data;
        Break;
      END If
    ELSE
      k ← 0;
    END If
  ELSE
    k ← 0;
  END IF
END WHILE

```

PSEUDOCODE 1: Pseudocode for the trigger signal detection of the receiving devices.

4. Experiments on and Analysis of the Data-Sharing Method among Multi-Smart Devices

This section introduces the development of the application for sharing data among multi-smart devices using the proposed method. To evaluate the performance of the proposed method, we conducted a comparative experiment between the existing Bump method and the proposed method in near data sharing. Moreover, we discuss the survey on and the result of the usability of the proposed method. Figure 8 shows the screen of the Multi-Bump application using the proposed method. The initial screen of the Multi-Bump application is shown in Figure 8(a), and the screen of the Multi-Bump application when the application receives the shared data is presented in Figure 8(b). Figure 8(a) shows the application's image preview of the shared data, the "Select File" button, the "Change Key" button for selecting the trigger signal and now the trigger signal value, and the "Start Data Share" toggle

button to start and end data sharing in order from the top. When the application initially runs, the preview shows "No Image" because it does not select any shared data. When the user selects a shared file, the preview shows a thumbnail image of the shared file. The base signals of the trigger signal use 19.0 kHz and 22.0 kHz within the 18 kHz to 22 kHz range. The range of the low-latency key value that the user can select is from 19.6 kHz to 21.4 kHz. The start frequency of the low-latency key value is 19.6, and this frequency is 600 Hz greater than 19.0 kHz as the base signal. Furthermore, the end frequency of the low-latency key value is 21.4 kHz, and this frequency is 600 Hz lower than 22.0 kHz as the base signal [28]. These frequencies are used to avoid interference between the base signals and the low-latency key signal. As shown in Figure 8(a), the Multi-Bump application can have 19 types of trigger signals. The low-latency key value shown in Figure 8(a) is 20.3 kHz (eight-channel).

If the receiving device wants to receive the shared data, it must set the same low-latency key value of the sending device. In Figure 8(b), the three position (①–③) states of the receiving device screen change when the shared data are received. Position ① as the preview for the shared data shows an activity indicator during the receiving of the shared data. Two buttons of position ② are changed from a usable state to a nonusable state to avoid the button control error during the receiving of the shared data. The progress bar of position ③ appears only when the receiving device is receiving the shared data. It disappears when the receiving device finishes receiving the shared data.

In the experiment on the performance evaluation of the Multi-Bump application, we conducted a test to determine the suitable length of a trigger signal and threshold α . Chung and Choo used 104 ms as the length of the trigger signal, and threshold α was 52 ms. Thus, we tested changing the length of the trigger signal to 52, 78, 104, 130, 156, 182, 208, and 234 ms and set threshold α to 50%. The experimental environment was an office laboratory where the noise level was maintained at less than 40 dB. The distance of the sending device from the receiving device was from 1 m to 5 m. We sent the trigger signal 100 times according to the length of the trigger signal and the distance of the devices. We used 19.6 kHz as the low-latency key value. iPhone 6 was used as the sending

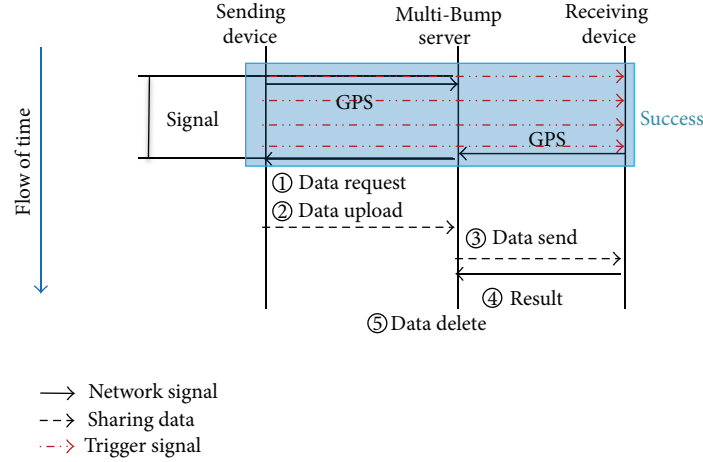


FIGURE 6: Flow of the proposed method using inaudible frequency for data sharing.

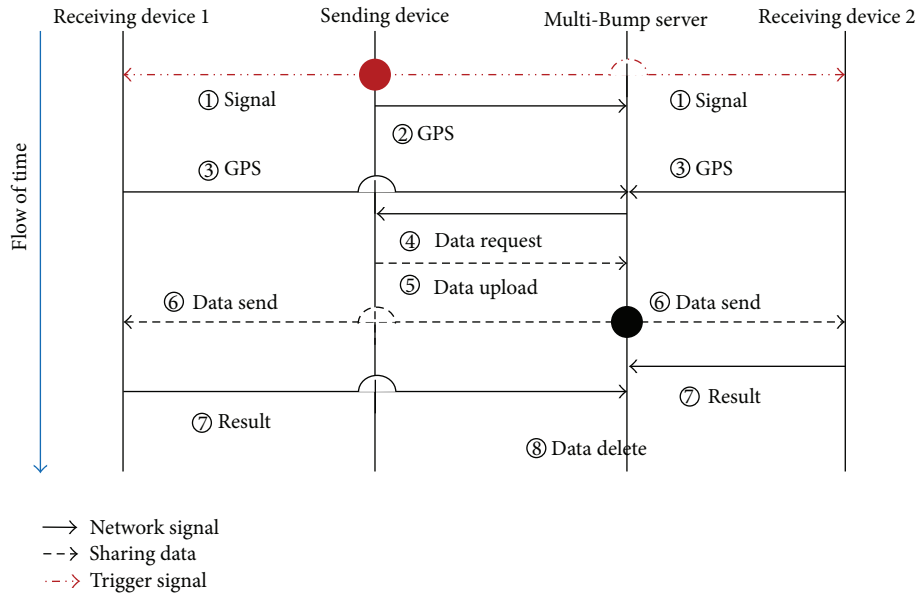


FIGURE 7: Data-sharing flow of one sending device and two receiving devices.

smart device and Galaxy S6 as the receiving smart device. Figure 9 shows the result according to the length of the trigger signal.

As shown in Figure 9(a), when the length of the trigger signal is short, such as 52 ms and 78 ms, the transmission accuracy is under 83%. As shown in Figures 9(a) and 9(b), when the length of the trigger signal is in the middle, such as 104, 130, and 156 ms, the transmission accuracy is over 97%. When the length of the trigger signal is long, such as 182, 208, and 234 ms, the transmission accuracy is over 99%. Thus, as the trigger signal length of 208 ms obtained the best accuracy in this test, we set the length of the trigger signal to 208 ms.

Next, to find the value of threshold α , which causes the high accuracy of the proposed method, we conducted a transmission test according to the distance between the smart devices with a changing threshold α . The experimental

environment was the same as that in the previous experiment, and the length of the trigger signal was 208 ms. The changes in threshold α are 104 ms (50% of 208 ms), 125 ms (60%), 146 ms (70%), 166 ms (80%), and 187 ms (90%). Figure 10 shows the result of the transmission accuracy according to the distance between the sending smart device and the receiving smart device with a changing threshold α . As shown in Figure 10, the transmission accuracy values according to the changing threshold α are 99.2% (α : 104 ms), 99.2% (α : 125 ms), 99.8% (α : 146 ms), 97.2% (α : 166 ms), and 97.2% (α : 187 ms). When threshold α is 104 ms, 125 ms, and 146 ms, the transmission accuracy is over 99%. However, when threshold α is 166 ms and 187 ms, the transmission accuracy is under 98%. Therefore, in our experiment, we set threshold α to 146 ms.

The Multi-Bump server was made up of an Intel Core i5-4690 CPU and 8 G RAM. Linux was used as the server OS.



FIGURE 8: Application example applied to the proposed method. (a) Screen used to launch application; (b) screen used to receive the shared data.

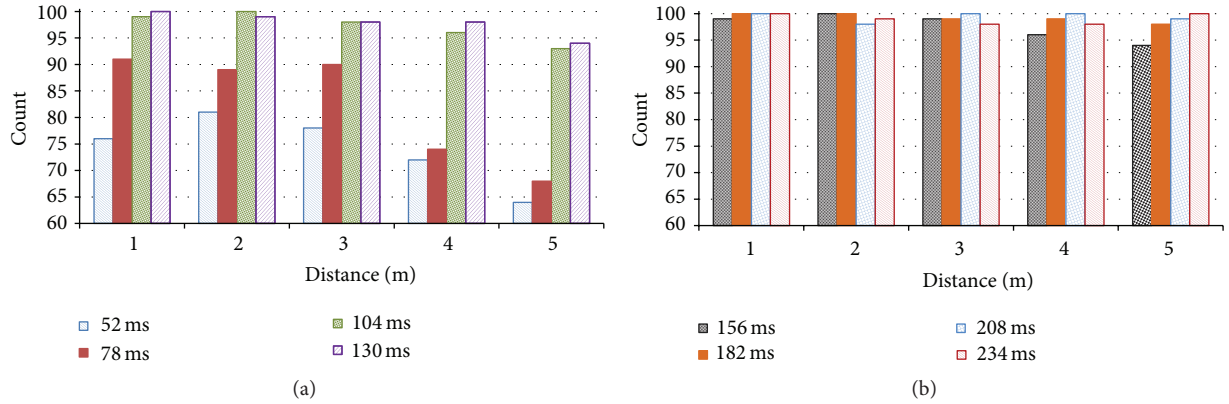


FIGURE 9: Transmission accuracy result according to the length of the trigger signal. (a) Result of using time duration from 52 ms to 130 ms; (b) result of using time duration from 156 ms to 234 ms.

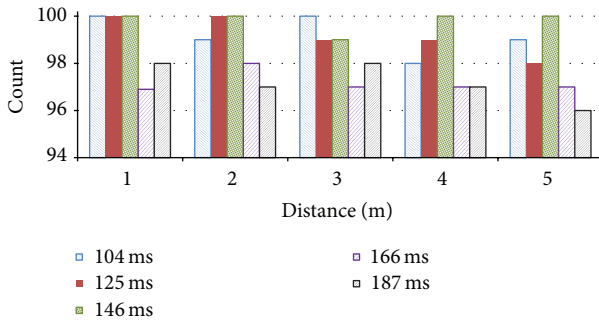


FIGURE 10: Result of the transmission accuracy according to the distance among smart devices with a changing threshold α .

Apache 1.3.41, PHP 5.2.6, and MySQL 5.0.51 were installed on the server. We used the POST function of PHP as the

transmission method between the server and the smart devices. The table schema of the database is presented in Table 1.

As shown in Table 1, no is the autoincrement index number, and regDate is the start time of the sending device for data sharing. signalCh is the low-latency key value when the sending device and the receiving device transmit the shared data. sGPSlo and sGPSla are the GPS information of the sending device, and receiveNo is the number of receiving smart devices joining the Multi-Bump data sharing. Count is the number of receiving smart devices that start to receive the shared data, and filename is the real filename of the shared data. To evaluate the performance of the proposed method, we conducted Chung's experiment. Chung and Choo did conduct an experiment where one smart device could control another smart device using a high frequency as the trigger signal [29]. The experiment environment had four conditions:

TABLE 1: Table schema of the database for data transmission between smart devices.

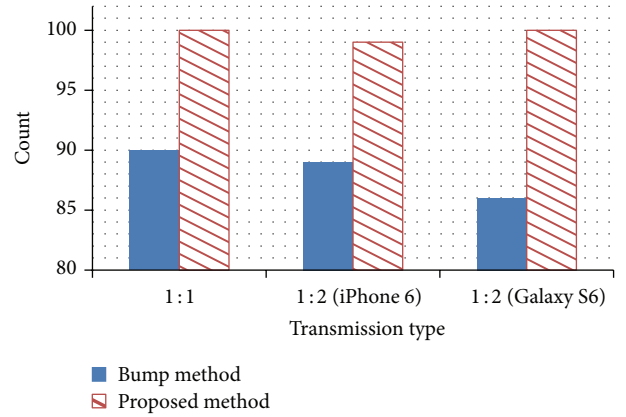
| Schema | Data value | Description |
|-----------|---------------------------------|---|
| no | int(11) unsigned auto_increment | Index number |
| regDate | int(11) unsigned | Data transmission time |
| signalCh | tinyint(3) | Using the channel number of low-latency key |
| sGPSlo | int(10) unsigned | GPS location of the data-sending device (longitude) |
| sGPSla | int(10) unsigned | GPS location of the data-sending device (latitude) |
| receiveNo | tinyint(2) | Number of devices that will receive the data |
| count | tinyint(2) | Number of devices that start the receiving of data |
| filename | varchar(40) | Real file name to be transmitted |

TABLE 2: Result of the performance evaluation of the proposed method according to the surrounding noise.

| Environment | Distance | | | | | | | Total |
|----------------|----------|-----|-----|-----|-----|-----|-----|-------|
| | 1 m | 2 m | 3 m | 4 m | 5 m | 6 m | 7 m | |
| Quiet indoors | 100 | 100 | 100 | 99 | 98 | 92 | 91 | 97.1% |
| Noisy indoors | 100 | 99 | 99 | 98 | 97 | 92 | 89 | 96.3% |
| Quiet outdoors | 100 | 100 | 99 | 99 | 95 | 94 | 90 | 96.7% |
| Noisy outdoors | 100 | 100 | 99 | 98 | 96 | 91 | 89 | 96.1% |

“quiet indoors,” “noisy indoors,” “quiet outdoors,” and “noisy outdoors.” In addition, the measurement distance for the near wireless-control was from 1 to 7 m, and each distance was tested 100 times. “Quiet” is 40 dB, such as typical household noise without conversation, and “noisy” is about 60 dB, the noise level of people talking in a typical conversational tone. “Quiet indoors” was in a laboratory that remained at about 40 dB, and “noisy indoors” was based on the same location, but with the addition of a water flow sound that was played from a PC speaker and with three subjects speaking freely. “Quiet outdoors” was a peaceful garden, which was maintained at about 40 dB, and “noisy outdoors” was a bus station where many vehicles passed; it was maintained at about 60 dB. Therefore, we conducted our experiment in a space similar to Chung’s experimental environment. The distance of the test was from 1 m to 7 m, and 100 attempts were made in each distance. A 2 MB photo image was used as the shared data, and 20.0 kHz (five-channel) was used as the low-latency key. Table 2 shows the experimental result of the data-sharing transmission. In Table 2, the transmission accuracy of the proposed method is similar to that of Chung’s experimental result. Chung’s experiment showed that the accuracy of the total average was 98.36% within 5 m and 96.09% within 7 m. The accuracy of the total average is 98.80% within 5 m and 96.57% within 7 m. Therefore, the proposed method is robust to the interference by unexpected sound signals.

Next, in Section 1, we wrote that the Bump application has not supported the Bump API and Bump server since January 31, 2014. Therefore, we developed an application and sharing server that work similarly to Bump API, and we conducted a comparative experiment on 1:1 and 1: N data sharing using the proposed method and the developed application,

FIGURE 11: Experimental results of the 1:1 and 1: N data sharing using the Bump application and the proposed method.

which works like Bump. The test for the 1:1 data sharing was performed in the quiet indoors. The distance between the sending device and the receiving device was 2 m, and 100 attempts were made using each method. We used one sending device and two receiving devices for the 1: N experiment similar to the previous experiment. The sending device was an iPhone 6, and the receiving devices were an iPhone 6 and a Galaxy S6. The distance between the sending device and the receiving devices was 2 m. To determine the usability of the proposed method, 20 college students were asked to join the experiment as participants. The students were divided into two groups. In the 1:1 data-sharing test, each group made 10 attempts using each method. In the 1: N data-sharing test, each group and the researcher made 10 attempts using each method. Figure 11 shows the result of the 1:1 and 1: N data-sharing experiments.

As shown in Figure 11, the Bump application (i.e., the existing method) succeeded 90 times, and the proposed method succeeded 100 times in the 1:1 data-sharing experiment. In the 1: N data-sharing experiment, the Bump method succeeded 89 times using the iPhone 6 and 86 times using the Galaxy S6 as the receiving device. The proposed method succeeded 99 times using the iPhone 6 and 100 times using the Galaxy S6. We considered that the reason why the accuracy of the Bump method was lower than that of the proposed method was that the time of the shaking action

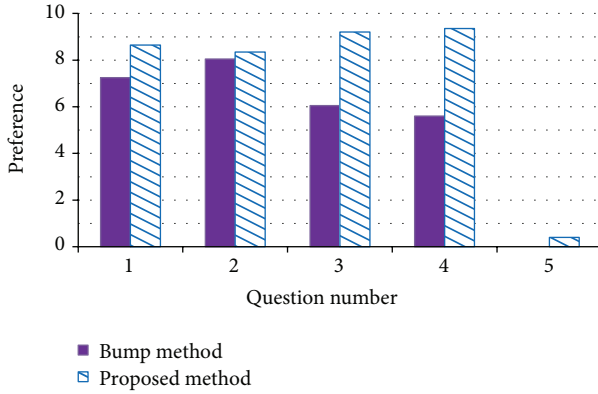


FIGURE 12: Survey results of the comparison between the Bump method and the proposed method.

for the data sharing sometimes did not exactly correspond. Moreover, in the 1: N data-sharing experiment, the sending device must perform the shaking action to each receiving device separately, as the Bump method can only perform 1:1 data sharing. Conversely, the proposed method has 99% high accuracy in both 1:1 and 1: N data sharing. This accuracy is similar to that of the previous experiment (four environments). Therefore, the proposed method is more accurate than the existing method in the 1:1 and 1: N data sharing.

Finally, we surveyed the 20 college students from the previous experiment using the following list to determine the usability of the Bump method and the proposed method. Each item is marked from one to ten points:

- (1) Convenience of the 1:1 data sharing.
- (2) Preference in the transmission time for the 1:1 data sharing.
- (3) Convenience of the 1: N data sharing.
- (4) Preference in the transmission time for the 1: N data sharing.
- (5) Strident disadvantage of the user in the high frequency as the trigger signal.

Figure 12 shows the survey results.

As shown in Figure 12, convenience of the 1:1 data sharing and preference in transmission time for the data sharing both gained 7 points. However, as the Bump method obtained 6 points in convenience of the 1: N data sharing and 5.5 points in preference in the transmission time for 1: N data sharing, we consider the Bump method to be unsuitable for 1: N data sharing. Conversely, the proposed method gained 9.2 points in convenience of the 1: N data sharing and 9.35 points in preference in transmission time for the 1: N data sharing. Therefore, we confirm that the proposed method is more suitable than the Bump method. As the points of the strident disadvantage of the user in high frequency as the trigger signal is under 0.5 points, the proposed method can be used in real life without any difficulty.

5. Conclusion

In this study, a method that uses inaudible high frequency for effectively sharing data among multi-smart devices at close range is proposed. The proposed method can solve the problems (e.g., Bluetooth pairing, different OS, and pretasks such as adding a friend) of the existing methods, and it is more accurate than the Bump method in 1:1 data sharing. Indeed, the proposed method can share data among multi-smart devices unlike the Bump method. Therefore, as the Multi-Bump method that uses inaudible high frequency does not need extra transmission modules, the proposed method can be applied to data sharing between a personal computer and smart devices as well as data sharing among multi-smart devices.

In our future research, we will examine direct data sharing among multi-smart devices without the Multi-Bump server by upgrading the trigger signal using inaudible high frequency. We will also study system development that can confirm people's focus on a specific space using a trigger signal and smart devices. Finally, in this paper, the sharing server did not consider security and privacy concerns related to the data stored on the Bump server. Thus, we had to handle the security and privacy problem. The developed application in this paper has a "Change Key" button that can change the high frequency of the trigger signal. If we apply the value of this button and the additional value to the application, we can use an encryption algorithm, such as the RSA method, and it should solve the security problem. Then, in Figure 3 of Section 3, when the data sharing between smart devices is concluded, the sharing server deletes the sharing data immediately. We think this function of the proposed method should solve the privacy concerns related to the data stored on the sharing server. Therefore, in our future research, we will study security and privacy concerns related to the data stored on the sharing server.

Conflict of Interests

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

Acknowledgment

This research was supported in part by Ministry of Education, under Basic Science Research Program (NRF-2013R1A1A2061478).

References

- [1] A. Iera, L. Militano, L. P. Romeo, and F. Scarcello, "Fair cost allocation in cellular-bluetooth cooperation scenarios," *IEEE Transactions on Wireless Communications*, vol. 10, no. 8, pp. 2566–2576, 2011.
- [2] H. Yoon and J. W. Kim, "Collaborative streaming-based media content sharing in WiFi-enabled home networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 4, pp. 2193–2200, 2010.
- [3] Apple Inc., <https://support.apple.com/en-us/HT204144>.

- [4] Z. Wang, Z. Xu, W. Xin, and Z. Chen, "Implementation and analysis of a practical NFC relay attack example," in *Proceedings of the 2nd International Conference on Instrumentation, Measurement, Computer, Communication and Control (IMCCC '12)*, pp. 143–146, IEEE, Harbin, China, December 2012.
- [5] M. Roland, "Software card emulation in NFC-enabled mobile phones: great advantage or security nightmare," in *Proceedings of the 4th International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use*, June 2012.
- [6] C.-H. Huang and S.-L. Chang, "Study on the feasibility of NFC P2P communication for nursing care daily work," *Journal of Computers*, vol. 24, no. 2, pp. 33–45, 2013.
- [7] S. Bouzeffrane, A. F. B. Mostefa, F. Houacine, and H. Cagnon, "Cloudlets authentication in NFC-based mobile computing," in *Proceedings of the 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud '14)*, pp. 267–272, April 2014.
- [8] C. Eunjeong, "Kakaotalk, a mobile social platform pioneer," *SERI Quarterly*, vol. 6, no. 1, pp. 63–69, 2013.
- [9] K. Church and R. De Oliveira, "What's up with whatsapp?: comparing mobile instant messaging behaviors with traditional SMS," in *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '13)*, pp. 352–361, Munich, Germany, August 2013.
- [10] J. D. Rachid, "Global communication apps," *Network Journal*, vol. 21, no. 1, p. 61, 2014.
- [11] S. L. Kane-Gill, J. P. Post, P. L. Smithburger, and A. L. Seybert, "'Bump': using a mobile app to enhance learning in simulation scenarios," *Simulation in Healthcare*, vol. 7, no. 5, pp. 326–327, 2012.
- [12] V. Filonenko, C. Cullen, and J. Carswell, "Indoor positioning for smartphones using asynchronous ultrasound trilateration," *ISPRS International Journal of Geo-Information*, vol. 2, no. 3, pp. 598–620, 2013.
- [13] B. Thiel, K. Kloch, and P. Lukowicz, "Sound-based proximity detection with mobile phones," in *Proceedings of the 3rd International Workshop on Sensing Applications on Mobile Phones (PhoneSense '12)*, p. 4, ACM, Toronto, Canada, November 2012.
- [14] G. Leventhall, "What is infrasound?" *Progress in Biophysics and Molecular Biology*, vol. 93, no. 1–3, pp. 130–137, 2007.
- [15] V. Coskun, B. Ozdenizci, and K. Ok, "A survey on near field communication (NFC) technology," *Wireless Personal Communications*, vol. 71, no. 3, pp. 2259–2294, 2013.
- [16] A. Kumar, A. Arora, and C. J. Islam, "Near field communication (NFC): an expertise primer," *Discovery*, vol. 2, no. 4, pp. 20–25, 2012.
- [17] M. H. Eldefrawy and M. K. Khan, "Banknote validation through an embedded RFID chip and an NFC-enabled smartphon," *Mathematical Problems in Engineering*, vol. 2015, Article ID 264514, 8 pages, 2015.
- [18] J. Constine, "Bump pay lets you PayPal someone with a tap, but only in-person," *TechCrunch Hot Topics*, March 2012.
- [19] R. Lai, *ING Direct's iOS App Adds 'Bump' Money Transfer Feature for Well-Heeled Posses*, Engadget, May 2011.
- [20] G. Tan, M. Lu, F. Jiang, K. Chen, X. Huang, and J. Wu, "Bumping: A bump-aided inertial navigation method for indoor vehicles using smartphones," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1670–1680, 2014.
- [21] Bump-api-ios, <https://github.com/bumpstech/bump-api-ios>.
- [22] T. Halevi, D. Ma, N. Saxena, and T. Xiang, "Secure proximity detection for NFC devices based on ambient sensor data," in *Computer Security—ESORICS 2012: 17th European Symposium on Research in Computer Security, Pisa, Italy, September 10–12, 2012. Proceedings*, vol. 7459 of *Lecture Notes in Computer Science*, pp. 379–396, Springer, Berlin, Germany, 2012.
- [23] A. Studer, T. Passaro, and L. Bauer, "Don't Bump, Shake on It: the exploitation of a popular accelerometer-based smart phone exchange and its secure replacement," in *Proceedings of the 27th Annual Computer Security Applications Conference (ACSAC '11)*, pp. 333–342, Austin, Tex, USA, December 2011.
- [24] J. Edwards, "Signal processing: the driving force behind smarter, safer, and more connected vehicles," *IEEE Signal Processing Magazine*, vol. 28, no. 5, pp. 8–13, 2011.
- [25] P. Sarah, "With its latest update, Bump's mobile app replaces USB flash drives," *Techcrunch*, <http://techcrunch.com/2013/02/14/with-its-latest-update-bumps-mobile-app-replaces-usb-flash-drives/>.
- [26] P. Bihler, P. Imhoff, and A. B. Cremers, "SmartGuide—a smart-phone museum guide with ultrasound control," *Procedia Computer Science*, vol. 5, pp. 586–592, 2011.
- [27] A. S. Nittala, X. D. Yang, E. Sharlin, S. Bateman, and S. Greenberg, "SonicData: broadcasting data via sound for smart-phones," Research Report 2014-1064-15, PRISM: University of Calgary Digital Repository, Calgary, Canada, 2014.
- [28] J. B. Kim, J. E. Song, and M. K. Lee, "Authentication of a smart phone user using audio frequency analysis," *Journal of the Korea Institute of Information Security and Cryptology*, vol. 22, no. 2, pp. 327–336, 2012.
- [29] M. B. Chung and H. S. Choo, "Near wireless-control technology between smart devices using inaudible high-frequencies," *Multimedia Tools and Applications*, vol. 74, no. 15, pp. 5955–5971, 2015.

