International Journal of Networking and Computing – www.ijnc.org, ISSN 2185-2847
Volume 9, Number 2, pages 318–338, July 2019

Evaluation of Ad-hoc Secure Device Pairing Method with Accelerometer and Camera Using Marker

Makoto Nagatomo
Security Research Center, Kanagawa Institute of Technology
Atsugi, Japan

Kentaro Aburada
Department of Computer Science and System Engineering, University of Miyazaki
Miyazaki, Japan

Naonobu Okazaki
Department of Computer Science and System Engineering, University of Miyazaki
Miyazaki, Japan

and

Mirang Park
Department of Information Network and Communication, Kanagawa Institute of Technology
Atsugi, Japan

Received: February 15, 2019
Revised: April 26, 2019
Accepted: June 4, 2016
Communicated by Susumu Matsumae

Abstract

Currently, devices with wireless technologies often communicate each other ad hoc. For example, a presenter wirelessly distributes ad-hoc meeting materials from a PC to mobile device in a meeting room. However, there is a problem of spoofing by an impersonator outside the room. Hence, devices must conduct secure pairing, which is exchange of key necessary for encrypting communication contents, before the communication. As a pairing method between devices, there are pairing methods using RSS from access point as features. However, RSS changes significantly due to environmental factors. On the other hand, there are the pairing methods which compare acceleration data from devices with displacement data of devices from camera of a server. However, these methods have problems that it is necessary to use infrared camera and difficult to recognize inclination of devices. Thus, these methods cannot perform accurate device pairing. Therefore, in this paper, we propose a method that perform pairing using devices' accelerometers and markers displayed on devices, and a camera of authentication server. This method performs pairing by calculating similarity between velocity data from acceleration data from devices and displacement data of the marker from camera after comparing marker sequence displayed on the device. This method has advantage that can detect devices' inclination by recognizing markers' inclination. We performed three types of experiments to confirm the similarity of displacement data and acceleration data, whether an impersonator outside camera range can perform pairing, and possibility of several devices pairing together. As a result, we founded that the larger the device's display is, the higher the similarity, the

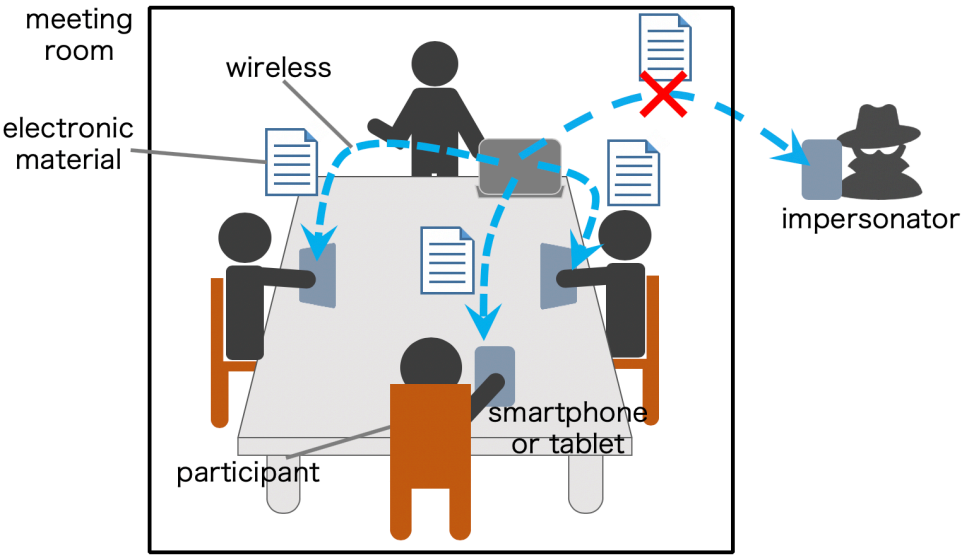


Figure 1: An example of ad-hoc device pairing

proposed method can distinguish legitimate user from impersonator outside camera range by average similarity, and three devices succeeded pairing at rate of 71.8%.

Keywords: device authentication, secure device pairing, camera-based authentication, device accelerometer

1 Introduction

Recently, along with the advance of wireless technologies, such as Wi-Fi, Bluetooth, and near-field communication (NFC), devices such as mobile device and IoT devices often use these wireless technologies to communicate with each other. These technologies have privacy vulnerabilities, such as eavesdropping and man-in-the-middle attack. Hence, it is necessary to establish authenticity between devices before wireless communication begin. We define secure device pairing as a process for establish secure wireless communication.

We classify device pairing into “long-term pairing” and “ad-hoc pairing”. Long-term pairing maintains the connection among devices long term, and the key for encrypting the content of communications can be prepared in advance. An example of long-term pairing is the connection between a Wi-Fi access point and smartphone. The access point has a key (security code) before a user is connected with it.

In contrast, ad-hoc device pairing maintains the connection only for a limited period. Fig. 1 shows an example of ad-hoc pairing, which is the wireless distribution of meeting materials from a PC to mobile devices with the small table (e.g. $3m \times 3m$ table) at a meeting . In this pairing, it is necessary to generate a pairing key ”on-the-spot”. The problem in this case is spoofing by a third party outside the room, who can obtain the meeting materials if secure pairing is not performed between devices in the room. When Wi-Fi or Bluetooth is used for it, the wireless communication may be eavesdropped. Therefore, our research goal is to propose the secure device pairing method, in which is easy to use for users, and it is possible to perform the pairing in a short range, such as the range of 3m $\times$ 3m.

Currently, many researches evaluate pairing methods using received signal strength (RSS) [1, 2]. The method [1] uses RSS between devices, and the method [2] uses RSS from several access points in order to judge whether the user in a room, but it is not possible to perform stable pairing in a single location because RSS changes significantly due to environmental factors, such as time and objects surrounded there.

On the other hand, the method using accelerometer is proposed [3]. In this method, a user shakes two devices together at the same time. However, it is possible that the impersonator can imitate the movement of the devices easily. In [4], the method using accelerometer and vibration of device is proposed. This method is secure because the vibration is not visible, but the pairing distance is very short.

There is the method that a device having function of emitting visible light conveys information to the device equipped with a camera [5, 6] in order to confirm whether the device is in front of it. However, the pairing distance of these methods are short (dozens of centimeters).

In addition, QR code is used to transmit data by reading it with a camera. For example, we can use QR code to transmit the key to encrypt transmission data as a pairing method. However, the impersonator outside the camera range can obtain QR code data to obtain private information by taking picture of it [7].

An alternative is proximity pairing, in which an infrared camera recognizes user's hand having holds a hand-held device [8, 9]. These methods calculate the similarity between hand movement data from infrared camera and acceleration data. However, this method requires a special infrared camera. Thus, method using a regular camera are proposed [10, 11]. These methods are tracking methods of objects with accelerometer by a camera. However, these methods is not secure because camera cannot recognize movements of the objects in detail.

In this paper, we propose a secure device pairing method that compares similarity between velocity data obtained from displacement data of marker displayed on the device and acceleration data from the device. In the proposed method, marker displayed on a device changes at an interval. The sequence of markers displayed on the device is used to distinguish a device from some devices by a camera in addition to calculation of the similarity of the velocity data. In addition, we perform three kinds of experiments to confirm the similarity displacement data and acceleration data, whether an eavesdropper outside the camera range can perform pairing, and perform three devices pairing together.

This paper is organized as follows. In Section 2, we review related works. Section 3 gives a system model and pairing procedure of the proposed method. In Section 4, we implement prototype application of the proposed method. In Section 5, we evaluate the similarity of marker sequence and velocity data of the proposed method. Finally, Section 6 concludes this research work.

2 Related Work

In this section, we introduce related work of our proposed device pairing method. Table 1 shows summary of related work.

2.1 Pairing Method Using Received Signal Strength(RSS)

Amigo [1] is an RSS-based method for pairing between devices. Amigo performs pairing by distinguish whether one device is near the other device using machine learning for the features of mean absolute difference of RSS, mean exponential of difference of RSS, and Euclidean difference of RSS vectors in order to detect the device. This method can detect an eavesdropper when the distance between the attacker and the legitimate devices is 3m.

In [2], device pairing in a room using RSS from some APs was proposed. This method performs pairing by distinguish whether devices are in the same room having size from 10 to 15m^2 using machine learning for features of RSS of beacon frames in 2.4-GHz band and 5-GHz bands. In addition, the set of APs of device is also one of characters. As a result, an accuracy of 96% was achieved in all cases.

However, RSS varies easily due to the environmental conditions around devices, so accuracy is unstable. For example, it is considered that the accuracy is low when the user moves the device or there are many people in the same room. Therefore, it is difficult to perform stable pairing with these methods. In addition, these methods have possibility that the impersonator through the wall can perform pairing because wireless overpass the wall and RSS is unstable.

Table 1: Summary of related work

Scheme	Application	Sensor	Method	Range	Evaluation	Limitation
Amigo[1]	pairing between devices	sensor receiving RSS	machine learning (features of RSS)	3m	0%(False Positive Rate)	RSS varies easily due to environmental conditions
Y. Agata et al.[2]	pairing between devices	sensor receiving beacon frame	machine learning (feature of RSS and set of APs)	room size from 10 to 15m^2	96% accuracy	RSS varies easily due to environmental conditions
D. Bichlder et al.[3]	pairing between devices	accelerometer	generation of common key from acceleration data	0m	70% accuracy (key of 13 bits)	impersonator can imitate movement of device
Vibreaker[4]	pairing between devices	accelerometer and vibration	convert vibration for 200ms into 1 bit	15cm	100% accuracy (transmission of 17 bits)	pairing distance is very short
N. Saxena et al.[5]	pairing between devices	light, camera	compare hash value of DH key	30cm	40 seconds of pairing time	pairing distance is short
Alex et al.[6]	transmission of packet	light, camera	convert one flash into 1bit	5-40cm	70% of packets were lost at 15cm	transmission distance is short
M. Rofouei[8]	pairing between device and display	accelerometer, infrared camera	compare movement of hand from camera with acceleration data	0.8-4m	92% accuracy	necessity of infrared camera
CrossMotion [9]	pairing between devices	accelerometer, infrared camera	compare movement of hand from camera with acceleration data	2.0m	99% accuracy	necessity of infrared camera
N. Maruhashi et al.[10]	object tracking method	accelerometer, camera	compare context of movement from camera and acceleration data	10m	94% accuracy	impersonator can imitate movement of legitimate user
S. Osamu et al. [11]	object tracking method	accelerometer, camera	compare movement of hand from camera with acceleration data	1-4m	0.8 of NCC (Normalized Cross Correlation)	impersonator can imitate movement of legitimate user

2.2 Pairing Method Using Accelerometer

When users shake devices together, the method [3] performs pairing by generating a common key necessary for encrypting transmission / reception data. This method generates a common key by combining the partial keys generated from the partial data sets of acceleration data. The pairing completes when the same key is generated between the devices. As a result, a common key of 13 bits is generated in success rate of about 70%. However, there is a possibility that the third party can generate the same key accidentally because this method uses only accelerometer.

Vibreaker [4] is a pairing method using accelerometer and vibration function. First, the user puts two devices close together. The one device then vibrates for 200ms in order to transmit 1 bit to the other device. The device that received vibration converts the acceleration data into 1 bit. From above procedure, this method performs pairing by transmitting 17 bits from the device to the device. This method has a problem that the pairing distance is very short because the device receiving vibration must be near the other device.

2.3 Pairing Method Using Visible Light and Camera

Methods using visible light are effective for device pairing within a room because visible light does not penetrate the wall. Saxena et al. [5] proposed a method for pairing between a device equipped with LED light and a device with a camera. This method first generates a common key in both devices by Diffie-Hellman (DH) key exchange method [12]. After that, the device transmits the hash value of the DH key to the other device through the camera using flashes of LED light. The device that receives the flash compares the received flash information with the hash value of the DH key. On the other hand, Alex et al. [6] propose the method that a device sends bits of packets by flashing LED light to a device. This method converts one flash of LED into 1bit. However, both of these methods have a limitation of the short pairing distance of dozens of centimeters.

Recently, QR code is used to transmit data frequently. Present smartphone such as iPhone and Android phones has function to read QR code. As an example of the pairing method, QR code can transmit a key to encrypt transmission data between devices. The user can perform pairing by just displaying QR code on a device's screen, and reading the QR code by a camera of another device. However, this method has possibility that the impersonator outside the camera range can obtain QR code data of legitimate user's device. In [7], an attacker successes to access private information by taking picture from QR code of a legitimate user and using it.

2.4 Pairing Method Using Accelerometer and Camera

Device pairing can be accomplished using an infrared camera (Kinect) [8, 9]. In [8], a PC equipped with Kinect and touch-screen associates the user's touch on the PC display with the user's smartphone by matching observed smartphone motion and motion transmitted by the smartphone. In [9], the method estimates image acceleration of a device by Kalman filter at each point of the image, and matches it with acceleration data from the device. These methods need a PC equipped with an infrared camera, and it does not directly detect the inclination of the smartphone.

Method in [10] is an object tracking of people equipped with accelerometer using regular camera. This method converts acceleration data from the device and movement data into the contexts, and compare two contexts. Three people are identified as a result of an experiment. Method in [11] identifies a moving object by a regular camera. It is assumed that the object has an accelerometer, and acceleration data is matched with the camera data using NCC (Normalized Cross-Correlation). When an object was 1-4m from camera, NCC was 0.8. However, if we apply these methods for device pairing methods, impersonator outside the range of camera can imitate users easily because these methods do not obtain inclination of the devices from camera. In addition, these methods are not secure because camera cannot recognize movements of the objects in detail, such as inclination of device.

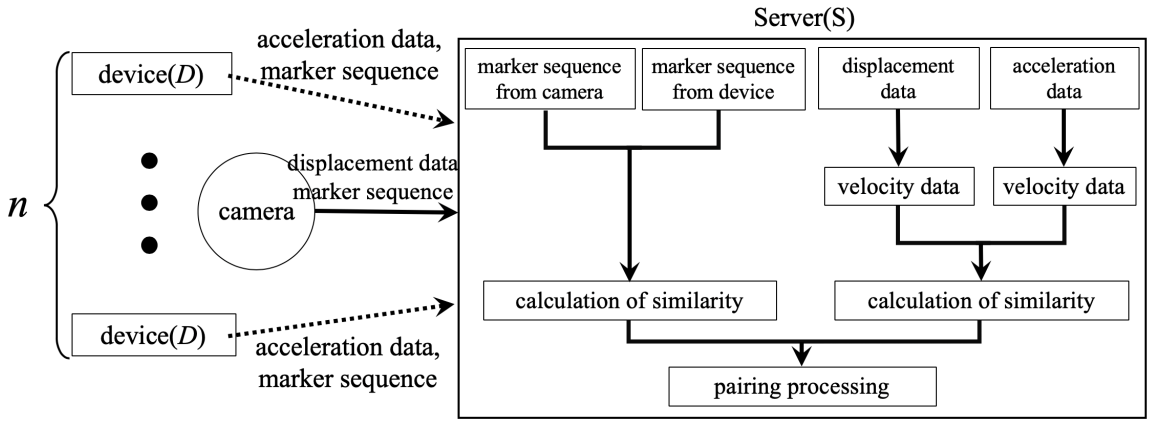


Figure 2: System model

3 Proposed Pairing Method between Server with Camera and Device Accelerometer

In this section, we propose a method for pairing between a device equipped with an accelerometer and a server equipped with a regular camera. The pairing of this method is performed by user's moving device in front of the camera. As judgement of permission of the pairing, this method calculates two similarity in order to perform authentication of a device in front of the camera. One is the similarity of marker sequence displayed on device's screen, that is, marker displayed on a device changes a certain interval in order to perform authentication of the device. The other is the similarity between velocity data obtained by integrating acceleration data and velocity data obtained by differentiating displacement data from camera. If the two similarities are higher than a certain value, this pairing succeeds. As a feature of the proposed method, we realize the detection of a device movement and its inclination by recognizing the motion and the inclination of the marker on the display of the device.

3.1 System Model and Flow of the Proposed Method

The system model of our proposed method is shown in Fig. 2. The components of this model are as follows.

- Authentication server (S)

  This is a server or other PC equipped with a camera, such as a web camera. The server calculates the similarity between acceleration data obtained wirelessly from the device and movement of the marker displayed on the device obtained from the camera.

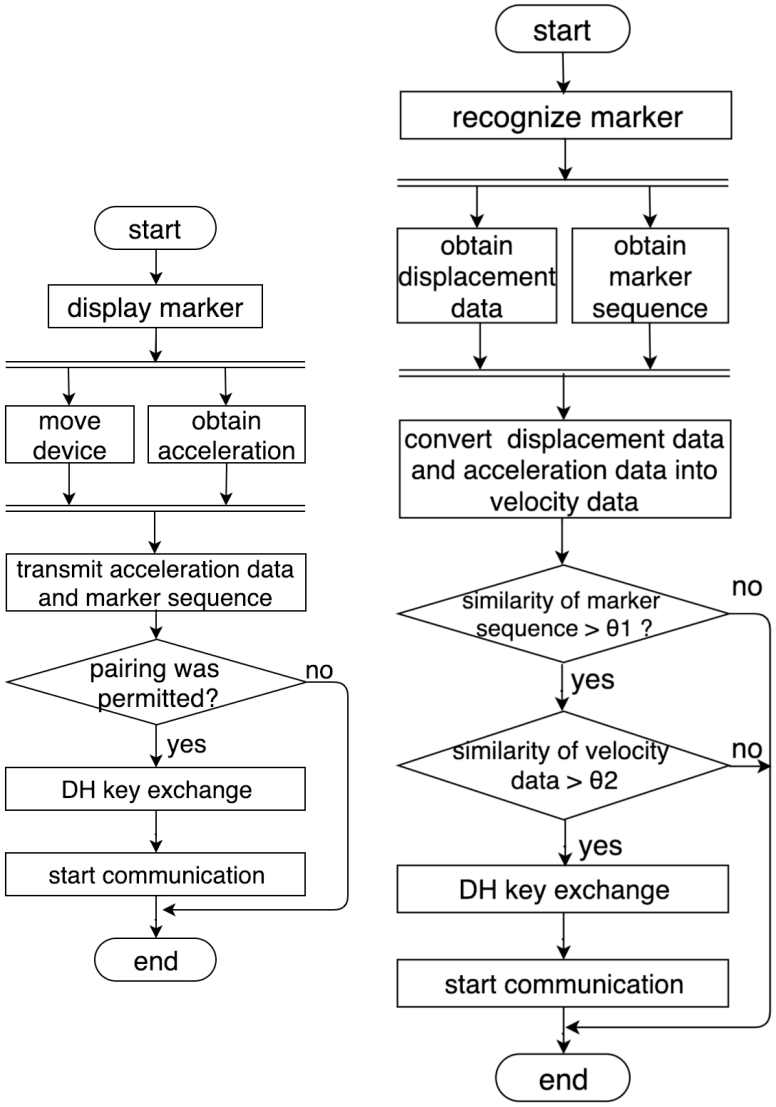
- Camera

  This is connected to the server and transmits the image information to it. This is not special camera such as infrared camera.

- Device (D)

  This is a device equipped with accelerometer, and displays the marker that the camera can recognize easily. This device transmits the 3-axis acceleration data to the server wirelessly at pairing. In this system model, it is assumed that some devices simultaneously perform the pairing with the server, hence there exists n devices.

Fig. 3 shows flow chart of the proposed method. Fig.3(a) shows flow chart of device's application. The device obtains acceleration data during moving device. After that, the device transmits the



(a) Flow chart of device's application (b) Flow chart of server's application

Figure 3: Flow chart of proposed method

acceleration data to the server. If the device receives the permission of pairing, performs DH key exchange with the server. Finally, the device starts to receive of materials from the server.

Server's application of flow chart is shown in Fig. 3(b). The server obtains displacement data of the marker and marker sequence from camera after recognizing the marker displayed on the device using the camera. After that, the server converts obtained displacement data and acceleration data into two velocity data by differentiating displacement data and integrating acceleration data. If the similarity of the marker sequence and two velocity data are certain value or more, the server transmits permission of pairing to the device, and performs DH key exchange with the device.

3.2 Pairing Procedure

We show the pairing procedure of our proposed method in Fig. 4 in the case of one device. The pairing procedure is as follows.

Step 1: *Display marker*

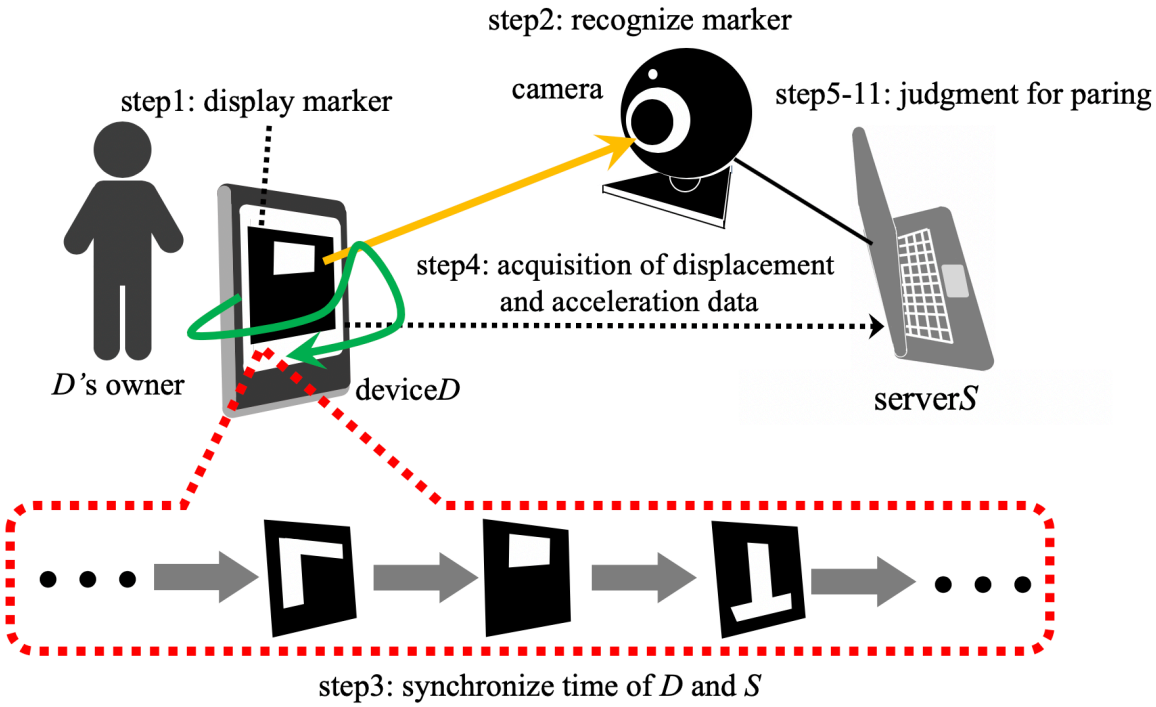


Figure 4: Pairing procedure

D displays a marker. This maker is a simple marker that the camera can detect easily. The marker displayed on the D's screen changes into another type of marker by a certain period of time (e.g. 0.5 second).

Step 2: *Recognize marker*

Server S recognizes the marker on the D's display through the camera.

Step 3: *Synchronization of time by changing marker*

This system synchronizes time of device D and time of server S. The first marker changing time is regarded as start time of this method in D and S.

Step 4: *Acquire displacement data and acceleration data*

The owner of D moves it arbitrarily, and D wirelessly transmits the set of acceleration data of x, y and z axis obtained from the device and its acquisition time to S. At the same time, S creates the marker displacement dataset β, which captures motion in the x and y axes. α and β are represented as follows.

$$\alpha = \{(a_i^x, a_i^y, a_i^z, m_i^\alpha, t_i^\alpha) | i \in \{1, ..., m\}\} \tag{1}$$

$$\beta = \{((x_j^1, y_j^1), (x_j^2, y_j^2), (x_j^3, y_j^3), (x_j^4, y_j^4), m_j^\beta, t_j^\beta) | j \in \{1, ..., n\}\} \tag{2}$$

where $a_i^x, a_i^y, a_i^z, m_i^\alpha (i \in \{1, ..., m\})$ represent acceleration along x, y and z axis, and type of displayed marker obtained at time t_i^α. Also, $(x_j^1, y_j^1), (x_j^2, y_j^2), (x_j^3, y_j^3), (x_j^4, y_j^4), m_j^\beta (j \in \{1, ..., n\})$ represent the coordinates of corners of the marker on the image and its type of marker at time t_j^β.

Step 5: *Calculate displacement data and correction of acceleration data*

Server S calculates marker's displacement and corrects acceleration data using marker's inclination, which can be obtained from four coordinates of the marker. Converted α' and β' are represent as follows.

$$\alpha' = \{(a_i'^x, a_i'^y, m_i^\alpha, t_i^\alpha) | i \in \{1, ..., m\}\} \tag{3}$$

$$\beta' = \{((x_j, y_j), m_j^\beta, t_j^\beta) | j \in \{1, ..., n\}\} \tag{4}$$

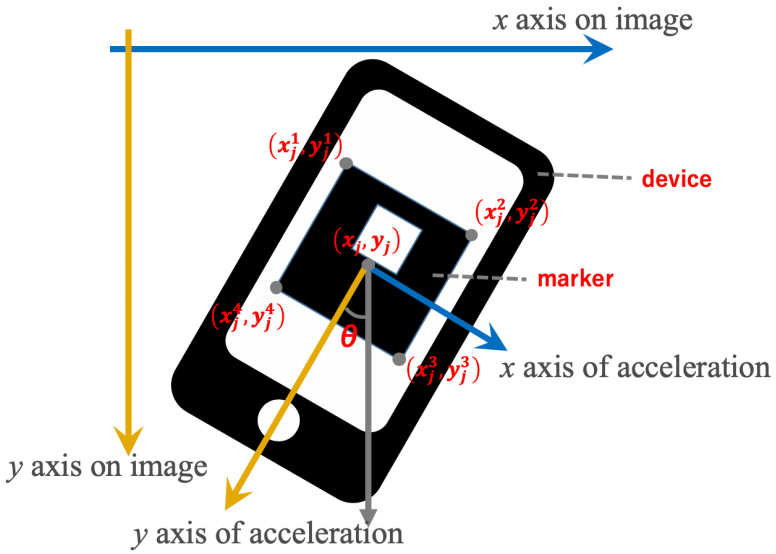


Figure 5: Correction of acceleration data by device's inclination

where $a_i'^x, a_i'^y$ and (x_j, y_j) are calculated as follows.

$$\begin{pmatrix} a_i'^x \\ a_i'^y \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} a_i^x \\ a_i^y \end{pmatrix} \tag{5}$$

$$x_j = \frac{x_j^1 + x_j^2 + x_j^3 + x_j^4}{4}, \quad y_j = \frac{y_j^1 + y_j^2 + y_j^3 + y_j^4}{4} \tag{6}$$

where θ is angle of device's inclination of the marker, and (x_j, y_j) is center of gravity of the marker (See Fig.5).

Step 6: *Eliminate noise*

Server S eliminates the noise by low-pass filter and gravitational acceleration by high-pass filter in acceleration data using Fast Fourier Transform (FFT).

Step 7: *Interpolation of displacement data*

The data on each x and y axis of the displacement data is approximated by three-dimensional spline interpolation. The function for the interpolation is as follows.

$$S_j(t) = a_j + b_j(t - t_j^\beta) + c_j(t - t_j^\beta)^2 + d_j(t - t_j^\beta)^3 \tag{7}$$

The five following conditions are used determine a_j, b_j, c_j, d_j.

1. $S_j(t_j^\beta) = w_j$
2. $S_j(t_{j+1}^\beta) = S_{j+1}(t_{j+1}^\beta) = w_{j+1}$
3. $S_j'(t_{j+1}^\beta) = S_{j+1}'(t_{j+1}^\beta)$
4. $S_j''(t_{j+1}^\beta) = S_{j+1}''(t_{j+1}^\beta)$
5. $S_0''(0) = S_{n-1}''(t_j^\beta) = 0$

where $w \in \{x, y\}$, $j \in \{1, ..., n-1\}$.

Step 8: *Convert to velocity data*

The S differentiates the motion data, and integrates the acceleration data to obtain the velocity data. In addition, we use function of spline interpolation in step 7 to displacement data to adjust the number of data. Each velocity datum α'' and β'' is represented as follows.

$$\alpha'' = \{(v_i^x, v_i^y, m_i^\alpha, t_i^\alpha) | i \in \{1, ..., m\}\} \tag{8}$$

$$\beta'' = \{(x_i', y_i', m_i^\beta, t_i^\beta) | i \in \{1, ..., m\}\} \tag{9}$$

Step 9: *Normalize data*

It is impossible to calculate the similarity between α'' and β'' directly because the units of two data are different. Therefore, we perform normalization for each axis by setting magnitude of a vector to 1. This vector is the data arranged in time order for each axis (the dimension of the vector is m). Each $\tilde{\alpha}, \tilde{\beta}$ obtained by normalizing α'' and β'' is as follows.

$$\tilde{\alpha} = \{(\tilde{v}_i^x, \tilde{v}_i^y, m_i^\alpha, t_i^\alpha) | i \in \{1, ..., m\}\} \tag{10}$$

$$\tilde{\beta} = \{(\tilde{x}_i, \tilde{y}_i, m_i^\beta, t_i^\beta) | i \in \{1, ..., m\}\} \tag{11}$$

where $\tilde{v}_i^w, \tilde{u}_i (w \in \{x, y\}, u \in \{x, y\}, i \in \{1, ..., m\})$ are calculated as follows.

$$\tilde{v}_w^i = v_i^w / \sum_{k=1}^{m} (v_k^w)^2, \quad \tilde{u}_i = u_i' / \sum_{k=1}^{m} u_k'^2, \tag{12}$$

and $\tilde{v}_i^w$ and $\tilde{u}_i$ take values from -1 to 1.

Step 10: *Calculate similarity of marker sequence*

Server S calculates the similarity of marker sequences obtained from S and D in $\tilde{\alpha}$ and $\tilde{\beta}$. If the similarity is over a certain value, this method proceeds step 11. If the similarity is under a value θ_1, this method finishes.

Step 11: *Calculate similarity between displacement data andacceleration data*

Server S calculates the similarity of velocity data between $\tilde{\alpha}$ and $\tilde{\beta}$. If the similarity is higher than the threshold θ_2, the pairing is accepted, and S and D generate a common key by DH key exchange. After that, the server transmits materials to the device.

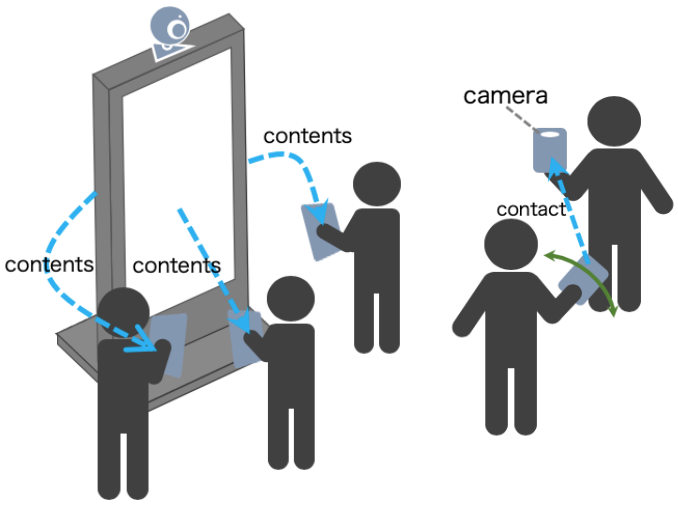
When the number of devices is more than one, acceleration data are separated by marker sequence. In addition, displacement data are separated by distance of displacement data. This method selects data having highest similarity between marker sequence and velocity data as the same device's data.

Benefits: DH key exchange is weak to man-in-the-middle attack, but our proposed method is resistant to man-in-the-middle attack because the proposed method can confirm that a device exists in front of a camera. In addition, this method does not need special camera such as infrared camera. Moreover, this method has longer pairing distance than QR code method because a marker of this method can be simpler than that of QR code method. Also, this method can perform high accuracy pairing because the inclination of the device can be detected. Hence, one-to-many device pairing is possible by reading markers on the displays of several devices.

Benefits of our proposed method compared with related work are as follows.

- Can separate users from impersonators through the wall
  RSS method [1, 2] has possibility that the impersonator through the wall success pairing because wireless overpass the wall and RSS is unstable due to environmental conditions. In our proposed method, an impersonator through the wall cannot perform pairing because the camera cannot recognize the impersonator through the wall.

- Not need special camera
  The methods [8, 9] need infrared camera to recognize movement of the hand having a device. Our proposed method does not need special camera such as infrared camera because of only recognizing the marker by a camera.



(a) Application for digital signage

(b) Application for exchanging contacts

Figure 6: Application of proposed method

- Can distinguish legitimate user from impersonator
  In [3], there is possibility that the impersonator outside the camera range can imitate the movement of the device of two users because this method generates common key from two acceleration data of two users directly. In [10, 11], there is also possibility that the impersonator can perform illegal pairing by imitating the user's movement because inclination of device is not detected or calculated. It is considered that our method can distinguish legitimate user's data from impersonator's data because the inclination of the device can be detected.

- More secure than QR code method
  By using QR code method, there is possibility that the impersonator outside the camera range obtains QR code of legitimate user [7]. On the other hand, our proposed method has resistance against impersonating legitimate user because the acceleration data and displacement data of the device are different depending on the person if different persons move devices in the same motion.

Use case: First use case is distribution of presenter's contents to people attending a meeting. Second use case is pairing between several device and digital signage (See Fig.6(a)). Users can operate digital signage together by displaying contents of digital signage on the devices, for example, in shopping. Third use case is several users' exchanging contacts together as if waving hands (See Fig. 6(b)).

3.3 Similarity Calculation of Marker Sequence

In our proposed method, server S calculates the sequences of markers obtained from camera and device in step 10 of pairing procedure. S calculates the similarity m_s as follows.

$$m_s = \frac{number\ that\ m_i^{\alpha} = m_i^{\beta}\ in\ \tilde{\alpha}\ and\ \tilde{\beta}}{m(number\ of\ set\ \tilde{\alpha}\ or\ \tilde{\beta})} \tag{13}$$

Fig. 7 shows an example of similarity calculation between marker sequences from the camera and the device. In this calculation, two marker sequences are the same, but the similarity is not 1.0. The reason why we employ this similarity calculation is because we can get stable similarity of marker sequence. If the camera misrecognizes the marker m_5^{β} as 1 when we use calculation method as comparing marker types of sequences (1,2,1,2,3) and (1,2,3), the similarity is low.

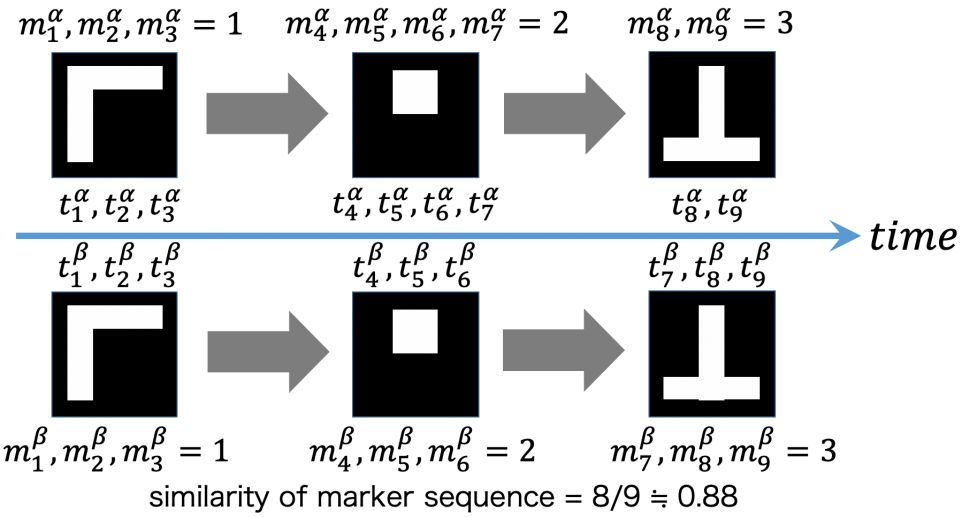


Figure 7: An example of similarity calculation of marker sequence

3.4 Similarity Calculation of Velocity data

In our method, the server S calculates the average of similarity of velocity in x and y axes as the similarity between $\tilde{\alpha}$ and $\tilde{\beta}$. We pick up four methods for calculating the similarity. Note that we do not consider calculation of similarity using z axis data of acceleration and inclination of the marker at this time.

1. Simple matching

   s_u is calculated as the average for summation of the difference of x and y axis data. The similarity is high when s_u is low. s_w is calculated as follows.

   $$s_u = \frac{1}{m}\sum_{k=1}^{m} |\tilde{v}_u^k - \tilde{u}_k|, \tag{14}$$

   where $u \in \{x, y\}$. The entire similarity is calculated by average similarity of x and y axis.

2. DP matching

   Function $g(m, m)$ is calculated from the following recurrence relation. The similarity is also high when $g(m, m)$ is low as well as simple matching.

   $$g(i,j) = min \begin{cases} g(i-1) + c(i,j) \\ g(i-1, j-1) + 2c(i,j), \\ g(i, j-1) + c(i,j) \end{cases}$$

   where $w \in \{x, y\}$, the cost function $c(i,j) = |\tilde{v}_i^w - \tilde{w}_j|/m$ and $g(0,0) = d(\tilde{v}_1^w, \tilde{w}_1) = c(0,0)$. The entire similarity is calculated by average similarity of x and y axis.

3. Correlation coefficient

   The similarities of x and y axis are calculated as following equation.

   $$r_w = \frac{\sum_{i=1}^{m}(\tilde{v}_i^w - \tilde{v}^w)(\tilde{w}_i - \tilde{w})}{\sqrt{(\sum_{i=1}^{m}(\tilde{v}_i^w - \tilde{v}^w)^2)(\sum_{i=1}^{m}(\tilde{w}_i - \tilde{w})^2)}}, \tag{15}$$

   where $w \in \{x, y\}$, $\tilde{v}^w = \sum_{k=1}^{m} \tilde{v}_k^w$, and $\tilde{w} = \sum_{k=1}^{m} \tilde{w}_k$. r_w takes a value from -1 to 1. It can be judged that there is a positive correlation when r_w is positive. When r_w is negative, there is a negative correlation. The entire similarity is calculated by average similarity of x and y axis.

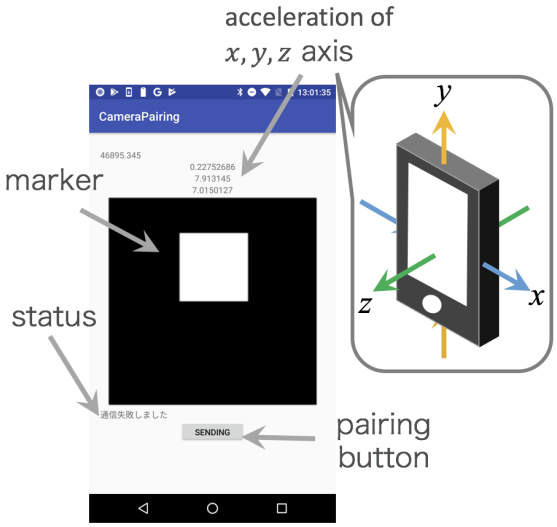


Figure 8: Application of the device

4. Jaccard index

   First, $\tilde{v}_i^w, \tilde{w}_i (i \in \{1, ..., m\}, w \in \{x, y\})$ are quantized to $\tilde{v}_i'^w, \tilde{w}_i' (i \in 1, ..., m)$ at 0.05 interval, then $\tilde{A} = \{(\tilde{v}_i'^w, i) | i \in 1, ..., m\}, \tilde{B} = \{(\tilde{w}_i', i) | i \in 1, ..., m\}$ is obtained. The distance between $\tilde{A}$ and $\tilde{B}$ is as follows:

$$similarity_w = \frac{|\tilde{A} \cap \tilde{B}|}{|\tilde{A} \cup \tilde{B}|}, \tag{16}$$

   where $similarity_w$ takes a value from 0 to 1, and the entire similarity is calculated by average similarity of x and y axis.

4 Implementation of Proposed Method

We implemented a prototype for evaluation experiment of the proposed method in Section 3. The server was a laptop PC (MacBook Pro 15-inch, 2017), and the device was mobile device of Nexus 5X(smartphone) and Nexus 7(tablet). The programs for the PC and device were developed using python3 and java in Android Studio. We used the camera and accelerometer built into the laptop and smartphone, respectively, by the manufacturers. Marker recognition was performed by ArUco [13], which is a library in OpenCV.

Fig. 8 shows the device application. The application starts the wireless communication with PC's application when the user starts the application. After that, a marker is displayed on the device's screen. When the user presses the pairing button, the application starts to obtain the acceleration data of x, y and z axis and the obtained time. The application transmits the set of the data when the button is pressed again.

Fig. 9 shows the PC application acquiring the device image. The video obtained from the camera is displayed during pairing. We set the size of the video as 640×360 pixels. When the user starts the device application, it initiates wireless communication with PC application. The application gets the 4 coordinates of the marker's corners. After receiving the acceleration data, this application calculates the similarity of marker sequence and the acceleration data and the displacement data.

We show an example of velocities calculated from the acceleration and displacement data in Fig. 10(a) and 10(b). The horizontal axis represents the time from the initiation of pairing. vel.X and vel.Y represent the obtained data along x and y axis. We find that the conversion is correct because the forms of the graphs are similar.

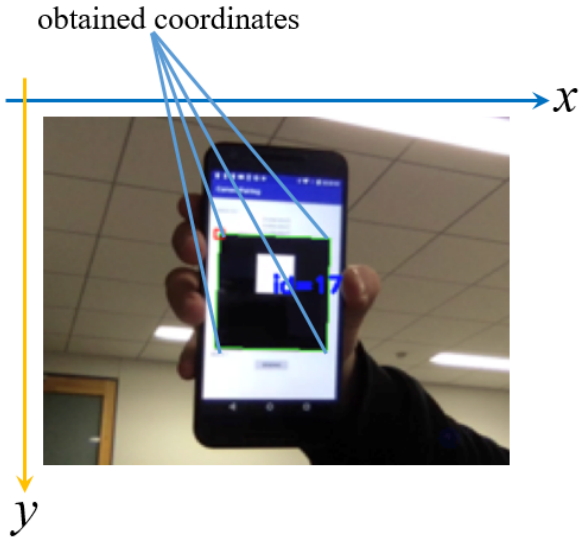
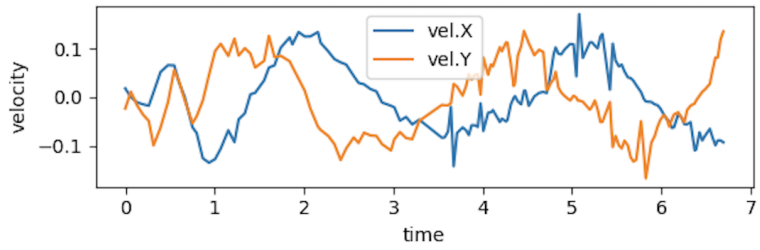
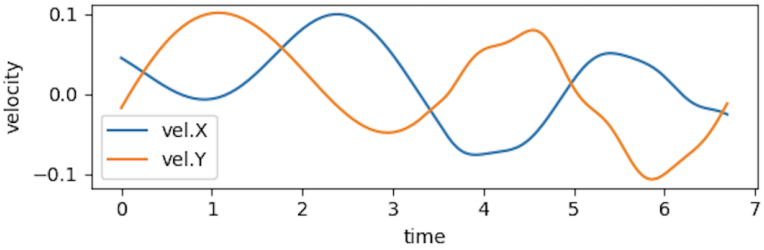


Figure 9: Application of the PC



(a) Velocity data from displacement data



(b) Velocity data from acceleration data

Figure 10: Velocity data from displacement data and acceleration data

5 Evaluation Experiment and Discussion

5.1 Experiment of Similarity Calculation of Velocity Data Using Only One Marker

In the proposed method in Section 3, it is possible that the similarity varies according to the device motion and the distance between the device and camera. In this section, we perform experiment to evaluate it by only one marker without using marker sequence using only Nexus 5X as a device.

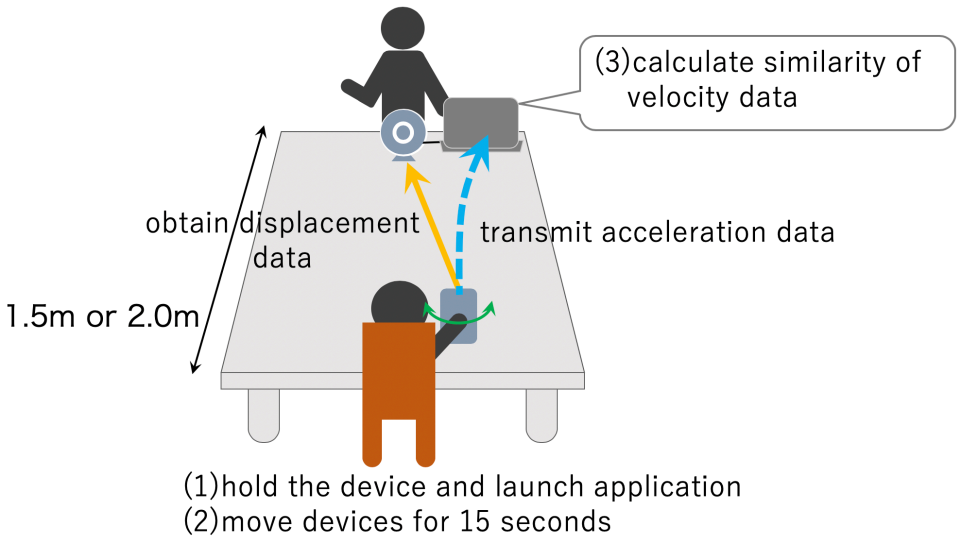


Figure 11: Procedure of experiment on confirmation of similarity of velocity data

5.1.1 Experiment for Similarity Confirmation of Velocity Data

We recruited 8 students belonging to Kanagawa Institute of Technology. The procedure of this experiment is as follows (See Fig. 11).

1. The subject holds the device (1.5 m, 2.0 m) away from the camera and launches the device application.
2. The subject holds the device so that it is oriented in a direction perpendicular to the floor and parallel to the laptop camera. The student moves the device along (the shape of a circle, ∞ symbol) for about 15 seconds.
3. The PC application calculates the similarity between the data obtained in step 2.
4. The student repeats steps 1-3 five times.

In this experimental environment, the maximum distance at which the laptop's internal camera could detect the marker was 2.0 m. Thus, we performed this experiment at 1.5 m and 2.0 m.

Table 2 shows the average of four calculation methods of the similarity. Simple matching and DP matching show high similarity as they are lower value. As a result, the similarity ∞ symbol motion was higher than that of circle motion at 1.5 m. However, the similarity of the circle was higher than that of ∞ motion at 2.0 m only for Jaccard index. On the other hand, the similarity using correlation coefficient was higher than that of Jaccard index. As a result, we found that the similarity of velocity data does not vary depending on motion and distance.

Table 2: Average similarity of velocity data for each method

motion, distance from camera	Simple matching	DP matching	correlation coefficient	Jaccard index
circle, 1.5 m	0.038	0.022	0.412	0.265
∞ symbol, 1.5 m	0.037	0.019	0.449	0.281
circle, 2.0 m	0.036	0.020	0.379	0.296
∞ symbol, 2.0 m	0.031	0.020	0.502	0.282

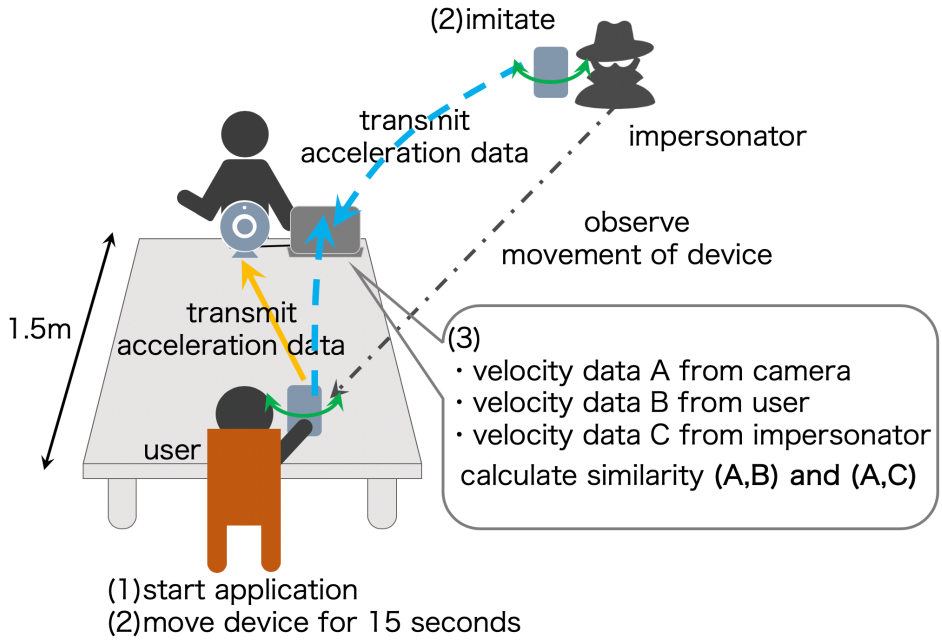


Figure 12: Procedure of experiment for impersonation

5.1.2 Experiment for Illegal Pairing

There is a possibility that the third party who is outside the range of the camera imitates the displacement of the legitimate user. If the similarity has variability, the similarity of the impersonating device can be higher than that of legitimate device. Therefore, we perform the confirmation experiment for considering it. Subjects are 6 students belonging to Kanagawa Institute of Technology participated in this experiment. We performed this experiment as following procedure (see Fig. 12).

1. Legitimate user holds the device 1.5 m away from server's camera. The user and impersonator start the device's application.

2. The user and the impersonator hold the their device so that it stands in a direction perpendicular to the floor. After that, the user moves the device along (the shape of circle, ∞ symbol) for about 15 seconds. Simultaneously, the impersonator imitates the displacement of user's device.

3. The server calculates the similarity between the displacement data obtained from step 2 and the acceleration data which is transmitted from both the user's device and the impersonator's device in step 2.

4. We repeated 1-3 five times.

Table 3 shows the average of similarities between legitimate users and impersonator. The result shows that the average similarities of legitimate user are higher than or equal to that of impersonator. Therefore, it is possible to determine a threshold between legitimate user and impersonator, but the standard deviations of the similarities were 0.022, 0.016, 0.175, 0.103, respectively. We found that this pairing method is unstable in this regard.

The rate that the similarity of legitimate user was higher than that of impersonator is shown in Table 4. The result shows that the rate using correlation coefficient was highest in the calculation methods of the similarity. Therefore, we use only correlation coefficient as the calculation of similarity from here.

Table 3: Average similarity for each method of legitimate user and impersonator

motion	simple matching		DP matching		correlation coefficient		Jaccard index	
	Legitimate user	Impersonator	Legitimate user	Impersonator	Legitimate user	Impersonator	Legitimate user	Impersonator
circle	0.039	0.045	0.023	0.028	0.67	0.53	0.31	0.26
∞ symbol	0.034	0.039	0.021	0.024	0.72	0.62	0.30	0.28
average	0.037	0.042	0.023	0.026	0.70	0.58	0.30	0.27

Table 4: Rate that similarity of legitimate user is higher than that of impersonator

motion	simple matching	DP matching	correlation coefficient	Jaccard index
circle	76%	36%	84%	68%
∞ symbol	56%	44%	80%	68%

5.2 Experiment for Proposed Method by Marker Variation

5.2.1 Experiment of Similarity Confirmation of Velocity Data and Marker Sequence

We perform a confirmation experiment of similarity calculation using marker variation version as well as section 5.1.1. We recruited 16 students belonging to Kanagawa Institute of Technology. The procedure of this experiment is as follows.

1. The subject holds the device 1.0 m away from the camera and launches the device application.
2. The subject holds the device for the laptop camera. The student moves the device along (the shape of a circle, ∞ symbol) for about 5 seconds.
3. The server calculates the similarity of marker sequence and velocity data.
4. The subject repeats steps 1-3 five times.

About the distance between the camera and the device, the maximum distance at which the camera could detect the marker was 1.0 m. Thus, we performed this experiment at 1.0 m.

Table 5 shows the average similarity of marker sequences. Similarity using Nexus 7(tablet) was higher than that using Nexus 5X(smartphone). Thus, we can find that the larger the display is, the larger the similarity is. Table 6 shows the average similarity of velocity data by correlation coefficient. The average similarity was larger than that in one marker version over 0.2.

Table 5: Average and standard deviation of similarity of marker sequence

device, motion	Nexus 5X, circle	Nexus 5X, ∞	Nexus 7, circle	Nexus 7, ∞	average
average similarity	0.717	0.688	0.855	0.854	0.778
standard deviation	0.248	0.217	0.213	0.199	0.232

Table 6: Average and standard deviation of similarity of velocity data

device, motion	Nexus 5X, circle	Nexus 5X, ∞	Nexus 7, circle	Nexus 7, ∞	average
average similarity	0.571	0.575	0.586	0.619	0.588
standard deviation	0.265	0.223	0.292	0.197	0.197

5.2.2 Experiment for Illegal Pairing Using Marker Changing Version

As well as section 5.1.2, we perform experiment whether the impersonator outside the range of camera can conduct pairing by imitating legitimate user's movement. Subjects are 12 university students. We performed this experiment as following procedure.

1. Legitimate user holds the device 1.0 m away from the PC's camera. The user and an impersonator start the device's application.

2. The user holds the device for the camera. After that, user moves the device along (the shape of circle, ∞ symbol) for 5 seconds. Simultaneously, impersonator imitates the displacement of the user's device.

3. The PC calculates the similarity of marker sequences and velocity data from user's device and impersonator's device.

4. We repeat 1-3 five times.

Table 7 shows the average similarities of marker sequence. The difference of the similarity between legitimate user and impersonator was very large. Therefore, it is almost possible to distinguish legitimate user from impersonator using only marker sequence.

Table 7: Average of similarity of marker sequence

motion, device	circle, Nexus 5X		∞, Nexus 5X		circle, Nexus 7		∞, Nexus 7	
	Legitimate user	Impersonator	Legitimate user	Impersonator	Legitimate user	Impersonator	Legitimate user	Impersonator
average similarity	0.647	0.02	0.776	0.013	0.664	0.023	0.664	0.019
standard deviation	0.377	0.03	0.232	0.029	0.266	0.039	0.276	0.03

We show the average similarities of velocity data between legitimate users and impersonators in Table 8. The result shows that the average similarities of legitimate user are higher than that of impersonator over 2.0. One marker version of the proposed method in section 5.1.2 has difference of average similarity under 2.0. In addition, it is possible to determine a threshold of similarity, but the standard deviations of the similarities were large. We found that this pairing method is unstable in this regard.

Table 9 shows the rate that the similarity of legitimate user is higher than that of impersonator. As a result, we can find the rate using (Nexus 5X, circle) was highest than that using Nexus 7 and the average rate was lower than the method using only one marker. This is because that the when the display of the device is large, impersonator can detect the legitimate user's movement easily.

Table 8: Average of similarity of velocity data

motion, device	circle, Nexus 5X		∞, Nexus 5X		circle, Nexus 7		∞, Nexus 7	
	Legitimate user	Impersonator	Legitimate user	Impersonator	Legitimate user	Impersonator	Legitimate user	Impersonator
average similarity	0.467	0.356	0.68	0.267	0.563	0.315	0.559	0.315
standard deviation	0.376	0.392	0.196	0.453	0.296	0.366	0.308	0.411

Table 9: Rate that similarity of legitimate user is higher than that of impersonator

device, motion	Nexus 5X, circle	Nexus 5X, ∞	Nexus 7, circle	Nexus 7, ∞	average
pairing success rate	81.8%	62.5%	75%	75%	73.4%

5.2.3 Pairing of Several Devices at the Same Time

We perform experiment whether several devices can be performed pairing together. Subjects are 12 university students. Devices used in this experiment are two Nexus 5Xs and a Nexus 7. Experimental procedure is as follows (See Fig. 13).

1. We make the team consisted of 3 people.
2. The 3 people launch devices' application, and move devices on the motion of circle or ∞.
3. The PC application calculates the similarity of marker sequence and velocity data for each device.
4. We repeat steps 2 and 3 five times.

Table 10 shows success rate that the proposed method separated three devices correctly. As a result, three devices' acceleration data and displacement data were separated correctly at rate of 71.8% on the average by comparing marker sequences of devices.

Table 11 shows the average similarity of velocity data and its standard deviation calculated by correlation coefficient. As a result, all average of the similarity and standard deviation were 0.461 and 0.299. Therefore, we find that the average similarity on this experiment is lower than that on first experiment for the similarity. This is because that the displacement data is not separated completely.

Table 10: Success rate of separating displacement data into three data

motion	circle	∞	average
average similarity	63.2%	80%	71.8%

6 Conclusion

In this paper, we proposed a ad-hoc secure device pairing method between a server equipped with a camera and a device equipped with the accelerometer. This method performs the pairing by

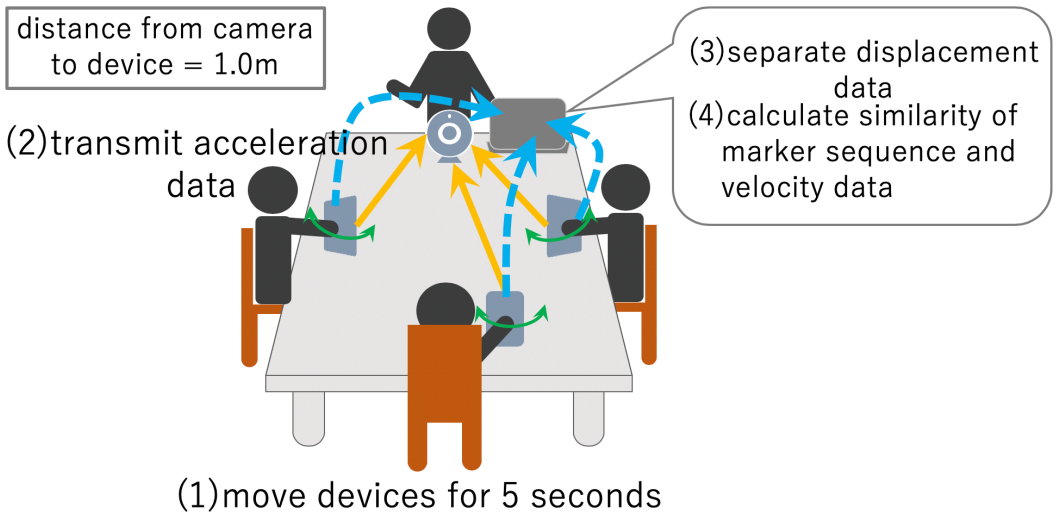


Figure 13: Experimental procedure of three devices pairing

Table 11: Average similarity of velocity data

motion	circle	∞	average
average similarity	0.492	0.431	0.461
standard deviation	0.28	0.316	0.299

calculating the similarity of marker sequence displayed on the device's screen, and the similarity of velocity data from camera and device's accelerometer.

We performed two experiments on the confirmation of similarity and illegal pairing using only one marker. In addition, we performed the three experiments on confirmation of similarity, illegal pairing can be performed, whether several devices can be distinguished using marker variation. As a result, the similarity was large as the size of device's display is large. Also, the average of similarity in the proposed method of marker variation version was higher than that of method using one marker. Moreover, we were able to separate the legitimate user and impersonator by the average similarity of velocity data. However, the stable pairing cannot be performed because the variation of the similarity was large. The result of the final experiment showed that some devices can be performed the pairing together, but the rate that the proposed method can distinguish three devices was 71.8%.

The future works are reduction of the variation for the similarity and raise of distinguished rate of some devices. For example, by using high resolution camera, we will examine the pairing distance and the similarities of marker sequence and velocity data.

References

[1] A. Varshavsky, A. Scannell, A. LaMarca, and E. de Lara, "Amigo: Proximity-based authentication of mobile devices," Proceedings of the 9th international conference on Ubiquitous computing (UbiComp 2007), pp. 253-270, 2007.

[2] Yugo Agata, Jihoon Hong, Tomoaki Ohtsuki, "Room-level proximity detection based on RSS of dual-band Wi-Fi signals," Proceedings of 2016 IEEE international conference on communications (ICC), 2016.

[3] D. Bichler, G. Stromberg, and M. Huemer, "Innovative Key Generation Approach to Encrypt Wireless Communication in Personal Area Networks," Proceedings of the 50th International Global Communications Conference, 2007.

[4] S. A. Anand and N. Saxena. "Vibreaker: Securing Vibrational Pairing with Deliberate Acoustic Noise," Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec '16). pp. 103-108, 2016.

[5] N. Saxena, J. Erik. Ekberg, and K. Kostianien, "Secure Device Pairing Based on a Visual Channel: Design and Usability Study," vol. 6, issue. 1, pp. 28-38, 2010.

[6] A. Duque, R. Stanica H. Rivano, and A. Desportes, "Unleashing the power of LED-to-camera communications for IoT devices," Proceedings of the 3rd Workshop on Visible Light Communication Systems, pp. 55-60, 2016.

[7] X. Bai, Z. Zhou, X. Wang, Z. Li, X. Mi, N. Zhang, T. Li, S. -M. Hu, and K. Zhang, "Understanding and mitigating synchronized token lifting and spending in mobile payment," 26th USENIX Security Symposium , 2017.

[8] M. Rofouei, A. D. Wilson, A. J. B. Brush, and S. Tansley, "Your Phone or Mine? Fusing Body, Touch and Device Sensing for Multi-User Device-Display Interaction," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 149-158, 2012. Camera View," Intelligent Robots and Systems, pp. 3872-3877 , 2008.

[9] A. D. Wilson and H. Benko, "CrossMotion: Fusing Device and Image Motion for User Identification, Tracking and Device Association," Proceedings of the 16th International Conference on Multimodal Interaction, pp. 216-223, 2014.

[10] N. Maruhashi, T. Terada, and M. Tsukamoto, "A method for identification of moving objects integrative use of a camera and accelerometers," Proceedings of the 27th annual ACM symposium on applied computing, pp. 1-6, 2012.

[11] S. Osamu, S. Kagami, and K. Hashimoto, "Identifying a Moving Object with an Accelerometer in a Proceedings of the 16th International Conference on Multimodal Interaction, pp. 216-223, 2014.

[12] W. Diffie and M. E. Hellman, "New directions in Cryptography," IEEE Transactions on Information Theory, vol. 22, no. 6, pp. 644-654, 1976.

[13] ArUco [online]. `https://docs.opencv.org/3.2.0/d5/dae/tutorial\_aruco\_detection.html.` Cited 20 June 2019.