[研究論文]

カメラと加速度センサーを利用したセキュアデバイスペアリング方式に関する研究

岡崎美蘭

情報ネットワーク・コミュニケーション学科

A Study on Secure Device Pairing Method using Camera and Acceleration Sensor

Mirang OKAZAKI

Abstract

RSS (Received signal strength) is used for proximate device authentication in wireless communications, but it changes significantly due to environmental factors. Thus, it is difficult to obtain reliably accurate device authentication with RSS. In this paper, we propose a method that performs authentication by pairing a PC having a common camera and a hand-held device having an accelerometer. In this method, the PC recognizes a marker on the display of the device and calculates the similarity between the displacement of the marker and the acceleration data of the device. We performed experiments to determine how the similarity changes according to the distance from a device to a camera and whether an eavesdropper outside the camera range can perform pairing. As a result, we found that it is possible to set a threshold for the similarity, but the standard deviation of the similarity was large, making the method unstable.

Keywords: device authentication, pairing, camera, acceleration sensor

1. はじめに

近年，スマートフォンやタブレット端末の普及と IoT（Internet of Things）機器の増加に伴い，これらのデバイスを用いた近接情報に基づくサービス(Proximity-based service)が注目されている．すなわち，あるデバイスから人やモノが近接にあるかどうかを検知し，それに応じたサービスを提供するものである．サービス例としては，周辺交通情報や天気予報などのニュース通知，クーポン券の配信や製品の関心度調査などのマーケティングでの活用がある．ここで，近接の定義は，アプリケーションによって数 mm から数 km まで様々である．近接情報はユーザの正確な位置情報を使わないので，位置情報が公開された場合のプライバシー侵害の問題はないが，位置情報の偽造によるサービスへの不正アクセスなどのセキュリティ上の問題がある．

近接情報技術では，環境音[1,2] や RFID(Radio Frequency IDentifier)[3] センサーが多く用いられている．しかし，これらの技術は誤検知が多く，検出範囲も局所的(1-2m)である．また，一般的な IoT 機器にはこれらのセンサーが含まれていない場合が多く，多様なサービスには適用できないという問題点がある．

そこで近年，Bluetooth[4]や Wi-Fi などの受信信号強度(RSS: Received Signal Strength)を用いた近接検出手法[5,6]が注目されている．Wi-Fi は，広範囲で受信することが可能なため，より広範囲で近接検出が可能だと考えられる．一方 Wi-Fi の受信信号強度を用いてサービスを提供するためには，アクセスポイント(AP)とスマートフォンが一時的にペアリングを行わなければならないという問題がある．これらを本研究では**デバイスペアリング**という．

デバイスペアリングには，提供するサービスによってデバイス認証が必要なペアリング(セキュアデバイスペアリング）とデバイス認証が不要なペアリングがある．すなわち，お店のクーポン券を配布する際には，できるだけ多くのユーザに配布したいので，相手のデバイスが受け取っても良い端末かどうか確認せずにクーポン券を配布する．しかし，お店のクーポン券をいくつかの割引率(例えば，10%引き/30%引き/50%引き）に分けて，会員別（一般会員，

シルバー会員，ゴールド会員）に配布しようとした場合，近くにいる端末がどの会員の端末であるかどうか確認する必要がある．このように，近くにいる特定の端末だけを通信相手として認識する仕組みを**セキュアデバイスペアリング**という．しかし，受信信号強度を使ってセキュアデバイスペアリングを行う際には，中間者攻撃（MITM: Middle-In-The-Middle attack）が可能である[7]．例えば，一般会員の端末がゴールド会員としてなりすまし，50%のクーポン券を獲得することが可能になる．

本研究では，上記のようなセキュアデバイスペアリングにおいて，第三者によるなりすましに強いセキュアデバイスペアリング手法を提案することを目的とする．

そこで本研究では，通常のカメラを備え付けた PC と加速度センサーを搭載したモバイル端末をペアリングする手法を提案する．本手法では，モバイル端末の画面上に表示したマーカーをカメラで認識し，ペアリングすべきモバイル端末の動きとみなす手法である．カメラが認識しやすいマーカーを読み取ることで端末の動きや傾きを検出しやすくなり，ペアリング精度の向上が期待できる．

以降，2. で関連研究を紹介し，3. で提案手法のシステムモデルとペアリング手順について述べる．4. で提案手法の実装と実験を行い，5. でその実験結果と考察を述べる．最後に 6. で全体のまとめと今後の課題について述べる．

2. 関連研究

2.1 受信信号強度(RSS)を用いた近接検出

Amigo[5]では，デバイス同士の無線通信の RSS の差の絶対値の平均，指数の平均，RSS ベクトルのユークリッド距離を特徴量とし，機械学習を用いてデバイス同士の近接検出をすることでペアリングを行う方式を提案している．複数の学習アルゴリズムを組み合わせることにより識別を向上させている．結果として，デバイス間が 5cm である場合，攻撃者（盗聴者）が 3m 以上離れている時に攻撃を検出できている．

縣ら[6]は，複数の AP からの RSS を用いた部屋単位でのデバイスペアリング手法を提案している．ここでは，複数の AP から送られる 2.4 GHz 帯と 5 GHz 帯の RSS の集合を特徴量とする．また，機械学習 k-Nearest Neighbor (KNN)法を用いて同一の 10m 四方の部屋にデバイスが存在するかどうかの識別を行なっている．結果として平均識別率が 99.3 %の精度が得られたが，シナリオによっては 88 %に低下している．これは RSS が周りの少しの環境の変化で値が変化するためである．

よって，RSS のみを用いたペアリング手法では部屋やパーテーションで区切られた空間内のみでペアリングを行うことは難しい．

2.2 加速度センサーを用いたペアリング手法

Smart-Its Friends[8]や Daniel ら[9]は，ペアリングを行いたい 2 つのデバイスを同時に振ることでペアリングを行う手法を提案している．ただし，両方のデバイスとも加速度センサーを搭載していると仮定する．特に，[8]では，得られた加速度データそのものからペアリング後のデータ送受信の暗号化に必要な共通鍵を生成する．共通鍵は，各々のデバイスについて，加速度データを複数に分割し，部分鍵を生成した後にそれらを合成することで得られる．各デバイスで同じ鍵が生成されればペアリング成功となる．結果として，平均 13 bit の共通鍵が約 70%の成功率で生成できたが，関係の無い第 3 者が同時にデバイスを振った際に同じ鍵が生成される可能性がある．

Vibreaker[10]は，加速度センサーに加えて，デバイスのバイブレーション機能を用いたペアリング手法である．まず 2 つのデバイスを密着させる．次に，片方のデバイスが 200ms の振動を行なったか行なってないかで 1 か 0 の 1bit の情報をもう片方のデバイスに送る手法である．振動を受けたデバイスは加速度センサーの変化により 1bit の情報をエンコードし，PIN コードを受け取る．それによってデバイス同士のペアリングが完了する．この手法では，通常の PIN に必要な 14 bit の情報量を送る際に，情報を送る合図の 3 bit を追加して送信する．よってペアリングに必要な時間は合計 3.4 s となる．問題点は，デバイス同士が振動を感知する距離内に存在する必要があるため，ペアリング距離が極端に短いことである．

2.3 カメラを用いたペアリング手法

部屋やパーテーションでペアリング可能範囲を区切る際に，電磁波の中でも壁を貫通しない可視光を用いて区切る手法[11, 12]が有効である．Nitesh ら[11]は，LED ライトなどを搭載したデバイスとカメラを搭載したデバイスをペアリングする手法を提案している．まず，ペアリングを行う 2 つのデバイスが無線を用いて DH 鍵交換方式[15]を使い，両方のデバイスに共通鍵を生成する．その後に片方のデバイスが視覚情報（LED ライトの点滅など）を用いてカメラを通じてもう片方のデバイスに DH 鍵のハッシュ値を送信する．受信デバイスは DH 鍵のハッシュ値を視覚情報で受け取った値と照合することでペアリングを行う．また，Alexis ら[12]による LED ライトの点滅で直接パケットの bit 情報を送る手法もある．しかし，どちらもペアリング可能距離が数十センチと短い制限がある．

また，赤外線カメラ(Kinect)を用いたデバイスペアリング手法も研究されている[13, 14]．山口ら[13]は，デバイス自体から得られた加速度データを無線で Kinect を搭載したサーバに送り，サーバがそれと手の動きのデータと照合することによりペアリングを実現している．この手法は，人が手にデバイスを持っていると仮定し，デバイスの動きを手の動きに代替している手法である．Mahsan ら[14]は，スマートフォンを持っている人とタッチパネルを操作する人を Kinect で認識する手法を提案している．しかし，どちらもデバイスの傾きを検出しておらず，さらに特殊なカメラである Kinect を用いる必要がある．

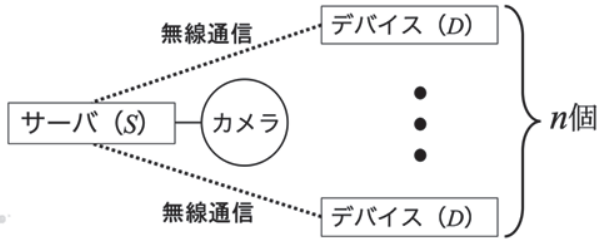

Fig.1 System Model

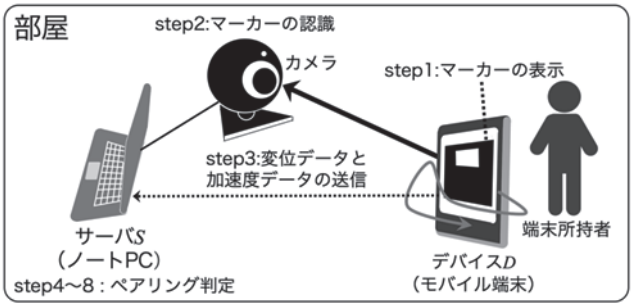

Fig.2 Pairing Procedure

そこで 本研究では，加速度センサーを搭載したモバイル端末と，通常のカメラを搭載した PC をペアリングする手法について検討する．デバイスの検出には，モバイル端末の画面上にマーカーを映し，カメラでマーカーを認識させることにより実現する．

3. カメラと加速度センサーを用いたペアリング方式

3.1 システムモデル

本論文で提案する方式のシステムモデルを図 1 に示す．以下そのシステムモデルの構成要素を示す．

(1) 認証サーバ（*S*）

カメラを搭載している，正規のデバイスを認証するサーバである．デバイスから無線通信で受信した加速度データと，カメラから得た画像データからマーカーの変位データを抽出し，それらの類似度を算出する．また，ペアリング結果をデバイスに送信する．

(2) カメラ

サーバと接続しており，デバイスから取得したマーカー画像情報をサーバに送信する．

(3) デバイス（*D*）

加速度センサーを内蔵したデバイスで，カメラで認識しやすいマーカーを画面に表示する．ペアリング時にデバイスの 3 軸の加速度データを計測し，サーバに送る．システムモデルでは，複数のデバイスがサーバと同時にペアリングを行うことを考え，n 個のデバイスがあるとした．

3.2 ペアリング手順

提案方式のペアリング手順を図 2 に示す．想定するサーバ *S* は，例えば WEB カメラを備え付けたノート PC で，デバイス *D* はモバイル端末とする．以下にペアリング手順を示す．

[step 1] マーカーの表示

D はマーカーを表示する．このマーカーはカメラが数メートル先でも認識可能な単純なマーカーとする．

[step 2] マーカーの認識

S はカメラを通じて *D* の画面上のマーカーを認識する．

[step 3] 変位データと加速度データの取得

D の所有者はマーカーをカメラに映したままデバイスを任意のモーションで動かし，その後に端末から得られた *x, y, z* 軸の加速度データとその取得時刻の組みの集合 α を無線でサーバ *S* に送信する．*S* はデバイスが動いている間，カメラ画像上のマーカーの変位データ x, y とその取得時間の組みの集合 β を取得する．α と β はそれぞれ次のように表される．

$$\alpha = \{(\alpha_1^x, \alpha_1^y, \alpha_1^z, t_1^\alpha), \ldots, (\alpha_m^x, \alpha_m^y, \alpha_m^z, t_m^\alpha)\} \quad (1)$$

$$\beta = \{(x_1, y_1, t_1^\beta), \ldots, (x_n, y_n, t_n^\beta)\} \quad (2)$$

ここで，各 $\alpha_i^x, \alpha_i^y, \alpha_i^z (i \in \{1,..,m\})$ は時刻 t_i^α で取得した *x, y, z* 軸の加速度を表し，$x_j, y_j (j \in \{1,..,n\})$ は時刻 t_j^β の画像上のマーカーの座標を表す．

[step 4] ノイズの除去

加速度データの重力加速度とノイズの除去を，高速フーリエ変換（FFT: Fast Fourier Transform）を用いてハイパス，ローパスフィルタで行う．

[step 5] データの補間

カメラデータの各 x 軸，y 軸のデータに対し，3 次元スプライン補間を用いて以下の関数で近似する．

$$S_j(t) = a_j + b_j\left(t - t_j^\beta\right) + c_j\left(t - t_j^\beta\right)^2 + d_j\left(t - t_j^\beta\right)^3 \quad (3)$$

また，a_j, b_j, c_j, d_j を決めるために，次の 5 つの条件が必要となる．

1. $S_j\left(t_j^\beta\right) = w_j$
2. $S_j\left(t_{j+1}^\beta\right) = S_{j+1}\left(t_{j+1}^\beta\right) = w_{j+1}$
3. $S'_j\left(t_{j+1}^\beta\right) = S'_{j+1}\left(t_{j+1}^\beta\right)$
4. $S''_j\left(t_{j+1}^\beta\right) = S''_{j+1}\left(t_{j+1}^\beta\right)$
5. $S''_0(0) = S''_{n-1}\left(t_j^\beta\right) = 0$

ただし，$w \in \{x,y\}, j \in \{1,\ldots,n-1\}$である．

[step 6] 速度データへの変換

S は[step 5]で得られたマーカーのデータを微分し，速度データ β' に変換する．また，[step4]で得られた加速度データを積分し，速度データ α' に変換する．各速度データ α' ， β' を以下のように表す．

$$\alpha' = \{(v_1^x, v_1^y, v_1^z, t_1^\alpha), \ldots, (v_m^x, v_m^y, v_m^z, t_m^\alpha)\} \quad (4)$$

$$\beta' = \{(x_1', y_1', t_i^\beta), \ldots, (x_m', y_m', t_m^\beta)\} \quad (5)$$

ただし，β'においては，[step 5]行なったスプライン補間の関数から加速度データの時刻と同じデータを補間し，データの個数を加速度データの個数と同じ m 個にした後に微分を行う．

[step 7] データの正規化

カメラから取得されたデータα'の単位は pixel/s であり，加速度センサーから得られた速度データβ'の単位は m/s であるため，そのまま類似度を算出しても妥当な結果を得ることができない．よって，各軸のデータ列に対して，取得時刻個数の次元（加速度データの取得個数 : m 次元）のベクトルとみなし，その大きさを 1 にすることで正規化を行う．各α', β'を正規化したデータ$\tilde{\alpha}$, $\tilde{\beta}$を以下の式で表す．

$$\tilde{\alpha} = \{(\tilde{v}_1^x, \tilde{v}_1^y, \tilde{v}_1^z, t_1^\alpha), \ldots, (\tilde{v}_m^x, \tilde{v}_m^y, \tilde{v}_m^z, t_m^\alpha)\} \quad (6)$$

$$\tilde{\beta} = \{(\tilde{x}_1, \tilde{y}_1, t_i^\beta), \ldots, (\tilde{x}_m, \tilde{y}_m, t_m^\alpha)\} \quad (7)$$

ただし，各$\tilde{v}_i^w, \tilde{u}_i$ $(w \in \{x,y,z\}, u \in \{x,y\}, i \in \{1,\ldots,m\})$を-1～1 の値を取るように以下の式で算出する．

$$\tilde{v}_w^i = v_w^i / \sum_{k=1}^{m} v_w^{k2} \quad (8)$$

$$\tilde{u}_i = u'_i / \sum_{k=1}^{m} u'^2_k \quad (9)$$

[step 8] 類似度の算出

上記式(6)と(7)から類似度を算出する．類似度の算出については次の節で述べる．閾値 θ 以上であればペアリングが成立し，DH の鍵交換方式で暗号化通信のための共通鍵を生成する．そうでなければペアリング不成立とする．

3.3 類似度算出方法

本論文では，3.2 節の[step 8]で類似度を計算する際に，正規化データ式(6)の x, y 軸のデータと式(7)の x, y 軸のデータの類似度を算出する．その後に x 軸，y 軸それぞれの類似度の平均を計算し，全体の類似度とする．類似度の計算には，以下の 4 種類を用いて類似度を算出した．なお，加速度データの z 軸データや，マーカーの傾きを考慮した類似度算出については，今後検討する予定である．

(1) 単純なマッチング

x, y 軸のデータの各取得時刻における値の差を全て足したのちに，その個数で割った平均を算出する．値が小さいほど類似度が高いことを意味する．類似度は以下の式で表す．

$$s_w = \frac{1}{m} \sum_{k=1}^{m} |\tilde{v}_k^w - \tilde{w}_k| \quad (10)$$

ただし，$w \in x, y$ である．

(2) DP マッチング

類似度を以下の漸化式から求める．$g(m,m)$ の値を算出し，それを類似度とする．算出された値は単純なマッチングと同様に値が小さいほど類似度が高いことを意味する．

$$g(i,j) = min \begin{cases} g(i-1) + c(i,j) \\ g(i-1,j-1) + 2c(i,j) \\ g(i,j-1) + c(i,j) \end{cases} \quad (11)$$

ただし，$w \in \{x,y\}$で，コスト関数 c を$c(i,j) = |\tilde{v}_i^w - \tilde{w}_j|/m$, $g(0,0) = d(\tilde{v}_1^w, \tilde{w}_1) = c(0,0)$とする．

(3) 相関係数

各 x 軸，y 軸での相関係数は，以下の式で算出する．

$$r_w = \frac{\sum_{i=1}^{m} (\tilde{v}_i^w - \bar{v}_w)(\tilde{w}_i - \bar{w})}{\sqrt{\left(\sum_{i=1}^{m} (\tilde{v}_i^w - \bar{v}_w)^2\right)\left(\sum_{i=1}^{m} (\tilde{w}_i - \bar{w})^2\right)}} \quad (12)$$

ただし，$w \in \{x,y\}, \bar{v}_w = \sum_{k=1}^{m} \tilde{v}_k^w, \bar{w} = \sum_{k=1}^{m} \tilde{w}_k$とする．$r_w$は-1 から 1 の値をとり，正の値が大きければ正の相関，負の値が大きければ負の相関があると判断できる．

(4) Jacard 係数

まず，各時刻のデータ$\tilde{v}_i^w, \tilde{w}_i (i \in \{1,..,m\})$ を 0.05 間隔で$\tilde{v}'^w_i, \tilde{w}'_i$と量子化し，その後に量子化した値とその時間の組みの集合 $\tilde{A} = \{(\tilde{v}'^w_i, t_i^\alpha) \mid i \in \{1,..,m\}\}$ ，$\tilde{B} = \{(\tilde{w}'_i, t_i^\beta) \mid i \in \{1,..,m\}\}$を考え，次式で集合の距離を定義する．ただし，$w \in \{x,y\}$である．

$$distance = \frac{|\tilde{A} \cap \tilde{B}|}{|\tilde{A} \cup \tilde{B}|} \quad (13)$$

$distance$ は 0 から 1 の値をとり，大きければ大きいほど類似度が高いことを示す．

4.　評価実験と考察

4.1 提案手法の実装

3 章で提案したペアリング手法の評価実験用のプロトタイプを実装した．開発環境として，サーバ側の開発言語は Python3 を使って開発した．マーカーの認識には OpenCV のライブラリ ArUco[16]を用い，デバイス側は Java 言語を用いて Android Studio 上で開発を行なった．実験機器はサーバをノート PC の MacBook Pro 15 inch 2017，デバイス側はモバイル端末 Nexus 5X をそれぞれ用いた．また，カメラと加速度データを取得する機器はそれぞれ内蔵されている機器を用いた．

図 3 に端末側のアプリケーションを示す．アプリケーションを立ち上げると画面に認識用のマーカーが表示され，PC 側のアプリケーションと，ルータを介した Wi-Fi の無線通信を開始する．また，アプリケーション起動中は，画面上に 5.5 cm×5.5 cm のマーカーを表示する．ペアリングボタンを押すと x, y, z 軸の加速度データとその取得時刻の取得が始まり，カメラに向けて端末を動かした後に再びペアリングボタンを押すと PC 側にそのデータを全て送信する．

図 4 に PC 側のアプリケーションを示す．アプリケーションを起動すると，カメラが取得した画像が表示される．カメラがマーカーを読み取ると，画像上に映っているマーカーの 4 つ角の座標が取得できる．本実験では，640×360 ピクセルの大きさの画像をカメラから取得するように設定した．また，端末側から加速度とその取得時刻のデータを取得すると，カメラから得た変位とその取得時刻のデータとの類似度を 3.3 節に従い算出する．

図 5，6 にペアリング時のデータの例を示す．横軸はペアリング処理開始からの時間を表しており，vel.X，vel.Y は x 軸と y 軸で取得されたデータを表す．速度データに変換する前のデータ個数はそれぞれ，変位データが 239 個，加速度データが 419 個であった．2 つのグラフを見比べると，外形が一致している箇所が複数見ることができ

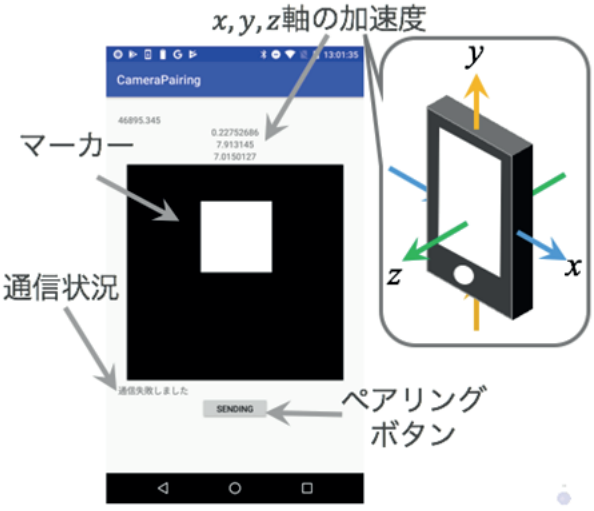

Fig.3　Application of the smartphone

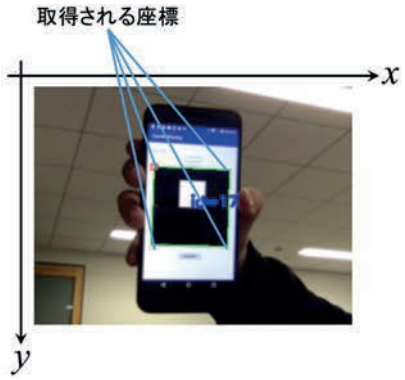

Fig.4　Application of the PC

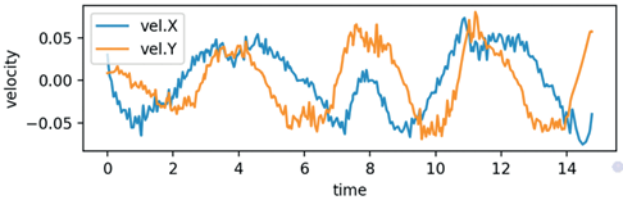

Fig.5　Velocity data converted from the movement data

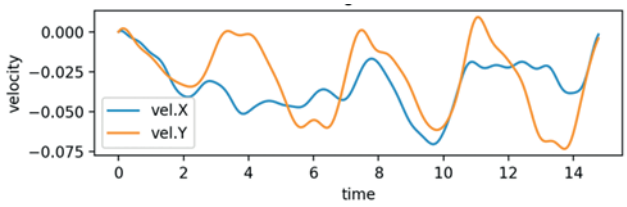

Fig.6　Velocity data converted from the acceleration data

るため，どちらも同じ速度データに変換できていることがわかる．しかし，x 軸の 2〜6 秒の間で形が大きく異なっている．これは，加速度センサーデータの方に大きくノイズが入ってしまったからであると考える．

4.2 実験 1 ：条件による類似度変化の確認実験

3 章で提案した手法において，デバイス（本実験ではモバイル端末である Nexus 5X）のカメラからの距離，モーションによって類似度が変化する可能性がある．よって，実験 1 として，その確認実験を以下の手順で行なった．

(1)　端末をカメラから（1.5 m，2.0 m）離す．
(2)　端末側のアプリケーションを起動し，端末側の画面上にマーカーを表示する．
(3)　端末を床に垂直な方向に立てるように持ち，カメラに平行にした状態で端末を(円，∞の字)の形に沿って約 15 秒動かす．
(4)　3.2 節の[step3]〜[step8]の手順により(3)で取得したデータから類似度を計算する．
(5)　(1)〜(4)を 5 回行う．

今回の実装環境では，PC がマーカーを読み取ることが

Table 1　Similarity for each Method

(モーション，カメラからの距離)	単純なマッチング	DP マッチング	相関係数	Jacard 係数
(円，1.5m)	0.038	0.022	0.412	0.265
(∞の字，1.5m)	0.037	0.019	0.449	0.281
(円，2.0m)	0.036	0.020	0.379	0.296
(∞の字，2.0m)	0.031	0.020	0.502	0.282

できる限界距離が約 2.0 m であった．よって，今回の実験は，読み取りができる限界距離から 50 cm 刻みの 1.5m と 2.0 m の距離で行なった．

表 1 に被験者 8 名の類似度の平均を示す．単純なマッチング，DP マッチングは数値が低いほど類似性があり，相関係数と Jacard 係数は数値が高いほど類似性があると判断される．結果として，1.5 m 地点での円と∞の字のモーションについては，∞の字のモーションの類似度が高いと算出された．しかし，2.0 m 地点では Jacard 係数のみ円のモーションの類似度が高いと算出された．円のモーションについて，相関係数のみ 2.0 m の類似性が低いと判断された．また，∞の字で DP マッチングのみが 2.0 m の方の類似度が低い結果となった．この結果から，近い距離では単純なモーションほど類似度が低く，マーカーが認識できる限界距離に近くなるに連れて類似度にばらつきが出てきたことが分かった．

4.3 実験２：不正なペアリングが可能かどうかの実験

提案手法において，カメラの範囲外にいるなりすまし者がペアリングを行う正規の端末の動きを真似して不正にペアリングが成功する可能性がある．もし，類似度にばらつきがあれば，正規の人よりなりすまし者の類似度が高くなる可能性がある．その不正なペアリングが可能かどうかを確かめるための実験を行った．なお，PC 側を Alice，正規の端末を持つ人を Bob，カメラに映る範囲外のなりすまし者を Eve とし，以下の手順で実験を行った（図 7）．

(1) Bob の端末を Alice の持つカメラから 1.5 m 離す．また，Eve はカメラに映る範囲外に移動する．
(2) Bob と Eve は端末を床に垂直な方向に立てるように持つ．Bob は端末の画面上のマーカーがカメラに映るようにし，カメラに平行にする．その後，端末を(円，∞の字)のモーションで約 15 秒動かす．動かしている間，Eve は Bob の動きを真似して端末を動かす．
(3) Alice は(2)で得た変位データと，それぞれ本人と盗聴者の加速度データの類似度を 3.2 節の［step3］～［step8］の手順により計算する．
(4) (1)～(3)を 5 回繰り返す．

以上の手順(2 つのモーションで 10 回)を，被験者 3 名が本人と盗聴者の両方を行える組み合わせで行なった．

Table 2　Similarities between legitimate user and Impersonator

モーション	単純なマッチング	DP マッチング	相関係数	Jacard 係数
円	76%	36%	84%	68%
∞の字	56%	44%	80%	68%

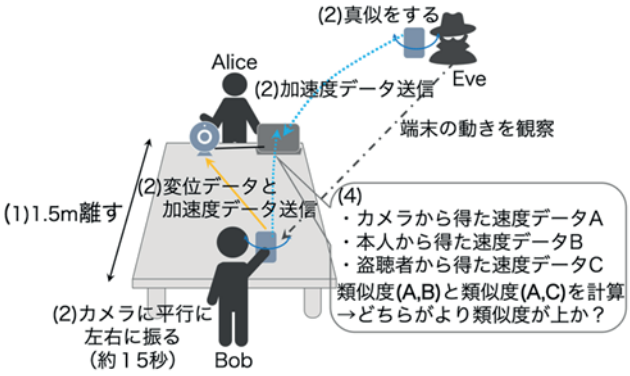

Fig.7 Experiment for illegal Pairing

表 2 になりすまし者より正規にペアリングを行う人の方が類似度が高いと判定された割合を示す．結果として，相関係数を用いれば，円，∞の字，両方どちらも含めて 84%, 80%と他の類似度算出方法と比べて高い結果となった．表 3 に本人と盗聴者の各類似度の平均を示す．どの類似度算出方法も平均を見れば閾値となる値を，0.39，0.23，0.65，0.29 と決めれば本人と盗聴者を識別できることが分かる．しかし，全体の標準偏差が 0.022，0.016，0.175，0.103 であったため，類似度が大きくぶれることも分かった．

5. まとめ

本研究では，カメラを備え付けたサーバとマーカーが表示できる画面と加速度センサーを備えたデバイスのペアリング方式の提案を行った．また，基礎的な実験として，x, y 軸方向のデータのみを用いた実証実験を行った．具体的には，距離とモーションによって類似度が変化するかどうかの実験と，カメラ範囲外のなりすまし者が正規な人のモーションを真似して不正なペアリングが出来るかどうかの実験を行った．

1 つ目の実験の結果から，カメラとマーカーの距離がマーカー認識限界距離（今回の実験では約 2.0 m）に近いほど類似度にばらつきが出ることが分かった．また，2 つの目の実験の結果から類似度の平均で本人と盗聴者を分けることができる閾値を確認できたが，類似度のばらつき（標準偏差）が大きくなることが分かったため，安定したペアリングができない結果となった．

Table 3　Each Similarity Average between legitimate user and Impersonator

モーション	単純なマッチング		DP マッチング		相関係数		Jacard 係数	
	正規な人	なりすまし者	正規な人	なりすまし者	正規な人	なりすまし者	正規な人	なりすまし者
円	0.039	0.045	0.023	0.028	0.67	0.53	0.31	0.26
∞の字	0.034	0.039	0.021	0.024	0.72	0.62	0.30	0.28
円，∞の字	0.037	0.042	0.023	0.026	0.70	0.58	0.30	0.27

今後は，3 次元のモーションやマーカーの傾きを類似度算出に用いることによって，類似度の標準偏差を減らす方法の検討をすることと，マーカーを複数同時に読み取ることによる一対複数のデバイスペアリングが可能か検証する必要がある．

参考文献

[1] D. Schurmann and S. Sigg, “Secure communication based on ambient audio,” *IEEE Transactions on Mobile Computing*, vol. 12, no. 2, pp. 358-370, 2013.

[2] N. Karapanos, C. Marforio, and C. Soriente, “Sound-proof : Usable two-factor authentication based on ambient sound,” in Proc. 24th *USENIX security symposium*, pp. 483-498, 2015.

[3] M. Bolic, M. Rostamian, and P. M. Djuric, “Proximity detection with RFID: A step toward the Internet of Things,” Pervasive Computing, IEEE, vol. 14, no. 2, pp. 70-76, 2015.

[4] S. Liu, Y. Jiang, and A. Striegel, “Face-to-face proximity estimation using bluetooth on smartphones,” IEEE Trans. Mobile Comput, vol. 13, no. 3, pp. 811-823, Apr. 2014.

[5] 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’07), pp. 253-270, 2007.

[6] 縣侑吾，洪志勲，大槻知明，“複数アクセスポイントからのデュアルバンド信号の受信信号強度に基づく部屋レベルの Proximity 検出,” 電子情報通信学会技術研究報告，vol. 115, no. 437, pp. 15-20, 2016.

[7] 藤井達也，小野貴継，金谷晴一，井上弘士，“受信信号強度を用いたデバイスに認証方式における攻撃可能条件の定式化,” 信学技報，vol. 116, no. 178, DC2016-19, pp. 15-22, 2016.

[8] L. E. Holmquist, F. Mattern, and B. Schiele, P. Alahuhta, M. Beigl, H. W. Gellersen, “Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts,” Proceedings of the 3rd international conference on Ubiquitous Computing (UbiComp’01), pp. 116-122, 2001.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 山口徳郎，立澤茂，野中雅人，“モバイル端末センサーと環境カメラを活用した端末ペアリング方式の提案,” 電子情報通信学会技術研究報告，vol. 112, no. 106, pp. 29-33, 2012.

[14] 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.

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

[16] ArUco, 入手先 <https://docs.opencv.org/3.2.0/d5/dae/tutorial_aruco_detection.html>(2018.5.12 参照)