# An Authentication Method Combining Spatiotemporal Information and Actions

Masateru Tsunoda*, Kohei Mitsui*, Kyohei Fushida*, Yasutaka Kamei*,
Masahide Nakamura**, Keita Goto* and Ken-ichi Matsumoto*

*Graduate School of Information Science, Nara Institute of Science and Technology,
Kansai Science City, 630-0192,
{masate-t, kohei-m, kyohei-f, yasuta-k, goto.keita01, matumoto}@is.naist.jp
**Graduate School of Engineering, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe, 657-8501,
masa-n@cs.kobe-u.ac.jp

## ABSTRACT

We propose a new authentication method combining actions and spatiotemporal information such as location, elapsed time, and travel distance. To be authenticated, a user performs certain actions at certain intervals defined with spatiotemporal information. For example, a user is authenticated when he/she pushes a button 5 times while going from point A to point B, pushes it 3 times while going from point B to point C, and pushes it 1 time while going from point C to point D. The proposed method is suitable for users who enter important places such as a secret data storage room, or a nuclear power plant. The advantage of the proposed method is high security. It is very difficult for others to recognize a moving user's actions and intervals defined with spatiotemporal information. We experimented to confirm the feasibility of the proposed method, and the results showed that the proposed method can be used for authentication.

***Keywords***: Authentication, Action, Spatiotemporal Information, Location, Speed

## 1 INTRODUCTION

User authentication by entering a password with a keyboard is a common method [13], however, it is not a very safe one because a password can be discovered by others through shoulder surfing. Therefore, better safety authentication methods have been proposed, such as smart card and biometrics authentication.

In smart card authentication [4] users are authenticated with a smart card that it is difficult to replicate. But there are some risks in that a smart card can be stolen. Although a user can disable his/her smart card when he/she notices it missing, the stolen smart card could be used for authentication before he/she notices the fact [7]. Biometrics authentication uses the fingerprint [6] or the iris [1] to authenticate users. Biometrics authentication is safe because it is difficult to replicate biometrics and impossible to steal them. However, biometrics authentication has some demerits. There is a possibility that a biometrics authentication system erroneously authenticates users with imitations [9]. Biometrics information cannot be disabled even if biometrics information leaks because it cannot be changed. Moreover, some biometrics authentication hardware requires installation space.

The defects of smart card authentication and biometrics authentication should be covered when high security is needed (e.g. authenticating the user who enters important places such as a secret data storage room, a military installation, or a nuclear power plant). The easy way to cover the defects is using two authentication methods (two factor authentication [14]). Two factor authentication usually uses an authentication method based on "What you know" in combination with an authentication method based on "What you have." Smart card authentication and biometrics authentication are authentication methods based on "What you have."

We propose a new authentication method that combines actions and spatiotemporal information such as location, time, and distance.[1] With the proposed method, a user performs specific actions on several sections that are defined by location, time, or distance. As shown in Figure 1, a user pushes the button 5 times between points A and B, 3 times between points B and C, and 1 time between points C and D to be authenticated.

The proposed method is an authentication method based on "What you know," and is suitable for two factor authentication when smart card authentication or biometrics authentication is used. The proposed method covers the defects of smart card authentication and biometrics authentication because it is robust against shoulder surfing, does not need a smart card that has the possibility of theft, and secret information for authentication can be changed. Our method can be used without other authentication methods. The proposed method is assumed to be used for authenticating a user who wants to be authenticated on their destination, and does not require any installation space around an entrance where authentication is needed.

The advantage of the proposed method is high security. Because the proposed method uses a combination of spatiotemporal information and actions as secret information for authentication, simple and quiet actions can be used for authentication. It is difficult for others to identify simple

[1] This research was partially supported by a Grant-in-Aid for 21st Century COE Research (NAIST-IS Ubiquitous Networked Media Computing).

and quiet actions of a moving user. The proposed method is more vulnerable to shoulder surfing than entering a password with a keyboard. When entering a password at one place is used for authentication, secret information of many users can be gotten by hidden camera. In the proposed method, an attacker should mark each user to get secret information. Even if the attacker did so, he/she would hardly get it because authentication actions are difficult to observe and he/she does not know which spatiotemporal information is used for authentication.

The proposed method can be easily extended and enhance security because a user can make complex secret information by changing the combination of spatiotemporal information and actions. In addition, the proposed method conceals itself and that enhances security. For example, when smart card authentication and the proposed method is used for authentication and others see that a smart card is used at an entrance, they would think the smart card is used for authentication but would not notice our method is used.

In what follows, section 2 defines the terms of the proposed authentication method and section 3 explains the structure of the system based on the proposed method. Section 4 reports an experiment to confirm the feasibility of the proposed method. Section 5 discusses usability and security of the proposed method and section 6 introduces related works. Section 7 concludes the paper with a summary and some future topics.

# 2 AUTHENTICATION METHOD COMBINING SPATIOTEMPORAL INFORMATION AND ACTIONS

## 2.1 Definition of Spatiotemporal Information

In this section, we define spatiotemporal information used in the proposed method. First of all, some basic terms are introduced for definition.

- **Location**: We define location $p_i$ as follows:

  $p_i := (x_i, y_i, z_i)$
  $x_i$:=latitude of the location
  $y_i$:=longitude of the location
  $z_i$:=altitude of the location. (1)

  The location is the place where a user is at one time, identified with a tuple of latitude, longitude and altitude.
- **Travel distance**: Travel distance is defined as a sum of the Euclidean distance from one point to another point. When a user goes from $p_{j1}$ to $p_{jn}$ by a series of $p_{j2}, \ldots, p_{jn\text{-}1}$, the definition of $dist_j$ is given as follows:

$$dist_j := \sum_{k=1}^{n-1} EuclideanDist(p_k, p_{k+1}) \ . \qquad (2)$$

  In the equation, $EuclideanDist(p_k, p_{k+1})$ is the Euclidean distance between Location $p_k$ and $p_{k+1}$.

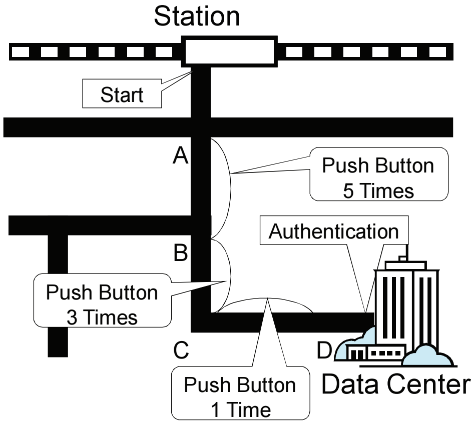

Figure 1: Example of authentication by the proposed method.

- **Elapsed time**: Elapsed time is defined as follows:

$$time_j := t_{jn} - t_{j1} \ . \qquad (3)$$

  In the equation, $t_{j1}$ is a time when a user is at $p_{j1}$, and $t_{jn}$ is a time when a user is at $p_{jn}$ in the same way.
- **Spatiotemporal information**: The definition of spatiotemporal information $spctmp_j$ is given as follows:

$$spctmp_j := \{(p_{j1}, \ldots, p_{jn}), time_j\} \ . \qquad (4)$$

## 2.2 Definition of the Authentication Interval

The three kinds of authentication interval $interval_j$ are defined based on spatiotemporal information $spctmp_j$ as follows.

- **Location-based authentication interval**: a location-based interval from $p_{j1}$ to location $p_{jn}$.
- **Time-based authentication interval**: a time-based interval from the time of $elapsedTime_j$ to the time of $elapsedTime_{j+1}$. $elapsedTime_j$ is defined as follows:

$$elapsedTime_j := \sum_{k=1}^{j} time_k \ . \qquad (5)$$

- **Distance-based authentication interval**: an interval based on total distance after starting authentication. $interval_j$ is defined as the difference of $totalDist_j$ and $totalDist_{j+1}$.

$$totalDist_j := \sum_{k=1}^{j} dist_k \ . \qquad (6)$$

## 2.3 Definition of the Authentication Point

The authentication point $point_j$ is defined as the authentication interval that turns down to a minimum. Three different kinds of definition of authentication point are given below.

- **Location-based authentication point**: $point_j$ is defined by location $p_j$.
- **Time-based authentication point**: $point_j$ is defined by $elapsedTime_j$.
- **Distance-based authentication point**: $point_j$ is defined by $totalDist_j$.

## 2.4 Definition of the Authentication Action

In the proposed method, a user repeatedly performs some actions on an authentication interval $interval_j$ or an authentication point $point_j$ to be authenticated. We call the actions "Authentication action $act_j$." Authentication action $act_j$ is defined by the combination of operation status values. The definition of operation status is shown below.

- **Operation status**: $sts_i$ is operation status at the time $t_i$. At least, operation status $sts_i$ must be a two-valued variable.
- **Authentication action**: The authentication action $act_j$ is given as follows:

$$act_j := (sts_{j1}, \ldots, sts_{ji}, \ldots, sts_{jm}) . \quad (7)$$

$(sts_{j1}, \ldots, sts_{ji}, \ldots, sts_{jm})$ means a sequence of operation statuses and an index $m$ is the number of operation statuses that the authentication action has. In the case of using the authentication point, the number of operation status $m$ is 1.

When the authentication interval is used, the number of authentication action patterns is larger than the number of operation status patterns. For example, if pushing a button is used for the authentication action, the values of $sts_{ji}$ are "pushing" or "not pushing." By using "$n$ times button pushing" as an authentication action, multiple patterns of authentication action can be defined. "$n$ times button pushing" can be used only when the authentication interval is used. On the other hand, in the case of using the authentication point, the number of elements included in $act_j$ is one. Hence, the number of authentication patterns is less than the number of the patterns of $sts_{ji}$.

## 2.5 Authenticating a User

As with the password authentication method, a user has to set the secret information that is necessary to be authenticated. In the proposed method, two different kinds of secret information are defined. Those are (1) the combination of sequences of authentication intervals ($interval_1$, … , $interval_j$, … , $interval_{ns}$) and authentication actions ($act_1$, … , $act_j$, … , $act_{ns}$), and (2) the combination of sequences of authentication points ($point_1$, … , $point_j$, … , $point_{np}$) and authentication actions ($act_1$, … , $act_j$, … , $act_{np}$). A user will be authenticated when his/her actions correspond with actions defined as secret information at all the intervals or the points.

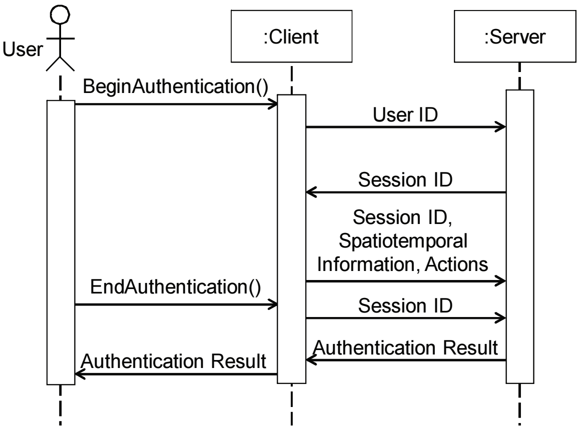

Figure 2: Authentication procedure using the proposed method.

## 2.6 Definitions of Terms

Terms used in this paper are defined as below.

- **Attacker**: a person who attempts authentication despite that he/she is not allowed authentication.
- **Client**: a terminal computer used by users when they attempt authentication. The client measures a user's spatiotemporal information and action status and sends them to a server.
- **Server**: a host computer that authenticates users. The server receives a user's spatiotemporal information and action status from a client and authenticates a user with them.
- **Authentication retry rate**: probability that a user retries authentication actions because of failure of authentication actions.
- **Authentication fault rate**: probability that users think their authentication actions are done correctly but the system recognizes they are incorrect and users fail to be authenticated.

# 3 THE STRUCTURE OF THE SYSTEM BASED ON THE PROPOSED METHOD

## 3.1 Authentication Procedure

This chapter explains the authentication procedure of the system based on the proposed method. We decided system specifications as follows.

- **Each user has a user ID.**
  Using the user ID, the system identifies the secret information of each user.

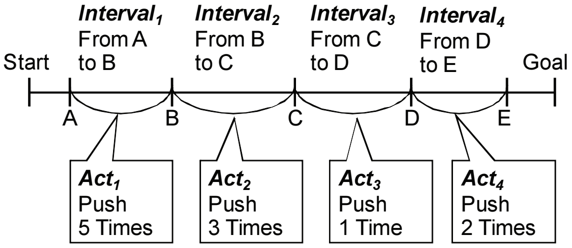

Figure 3: Example of secret information using location and pushing a button.

- **Make session ID on each authentication.** Without a session ID, if an attacker knows the user ID, the attacker can be authenticated when a user passes the last authentication interval (or point) and not arrive at the destination.
- **The client does not have any secret information for authentication.** This is to prevent an attacker from stealing the client and knowing secret information.

Following the specifications, we designed the authentication procedure as follows.

**(Step 1)** The client sends the user ID to the server.

**(Step 2)** The client accepts the session ID from the server.

**(Step 3)** The client sends the user's spatiotemporal information and action status with the session ID to the server.

**(Step 4)** When the user arrives at the destination, the client sends the session ID to finish authentication.

Figure 2 shows the authentication procedure based on the proposed method. To avoid communications between a server and a client being intercepted, data in steps 1 to 4 should be encrypted. In step 1, if a client device has a unique ID, it can be used instead of a user ID.

## 3.2 Settings of the Authentication Action

The proposed method adopts actions whose operation status $sts_{ji}$ has at least two variations. We proposed two authentication actions described below.

**Authentication action based on button status**

We apply button operations to an authentication action. It is difficult for an attacker to identify whether a button is pushed or not. To use the button operations, a small device like a mouse button, or buttons on a PDA or mobile phone are needed. When $n$ buttons are used for authentication, operation status $sts_{ji}$ has $2^n$ patterns. For example, when $n$ is 2, $sts_{ji}$ has 4 patterns like "button 1 and button 2 are pushed", "button 1 is pushed and button 2 is not pushed", "button 1 is not pushed and button 2 is pushed", "button 1 and button 2 are not pushed". Using authentication intervals, operation statuses $sts_{ji}$ like "push button $n$ times" or "push button for $n$ seconds" can be set.

**Authentication action based on mobile speed**

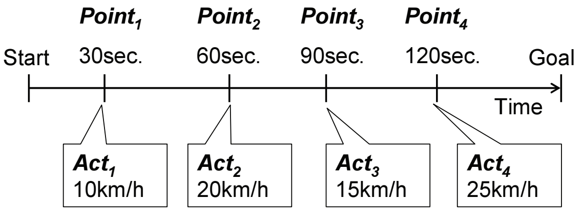

Figure 4: Example of secret information using speed and time.

We use mobile speed for setting $sts_{ji}$. Mobile speed is measured by the wheel revolution of a car or a two-wheeled vehicle, or calculated with the succession of motion data. With the combination of mobile speed and authentication points, the operation status such as "passing through an authentication point $point_j$ at a speed of $v_j$ kilometer per hour (km/h)" or "passing through an authentication point $point_j$ over $v_{jmin}$ km/h and at up to $v_{jmax}$ km/h" can be set.

Because mobile speed is changed not discretely but continuously, the combination of mobile speed and the authentication interval lowers the security of authentication. If "a user moves over $v_{j1}$km/h speed and at up to $v_{j2}$ km/h speed on an authentication interval $interval_j$" were set as an authentication action, an attacker would be authenticated successfully, changing mobile speed from lower speed to higher speed gradually.

When combined with authentication intervals, actions whose operation status cannot change discretely should be used with actions whose operation status can change discretely (e.g. an action of pushing a button). For example, with combining pushing button and mobile speed, "a user should move at 10km/h when pushing the button the first time, and at 15km/h when pushing the button a second time" can be set as an authentication action.

In this paper, when "passing through an authentication point $point_j$ at a speed of $v_j$km/h" is set as an authentication action, we name $v_j$ as **appointed speed**, and the user's actual passing speed at $point_j$ as **passing speed**. Mobile speed of a pedestrian is not appropriate for the authentication action because it has a small speed range.

Various actions can be applied as the authentication action. For example, actions based on rotation speed or angle are applied as an authentication action. Moreover, invisible operation status of actions like "holding breath, inhaling, or exhaling" can be applied as an authentication action using a sensor. However, it is not appropriate for the proposed method to use unusual and noticeable actions like the tracing of a PDA in the air [11] because that gives clues of authentication intervals or points to the attacker.

## 3.3 Examples of Secret Information Settings

This chapter illustrates examples of secret information settings (combination of authentication intervals and actions, or authentication points and actions), and a user's behavior to be authenticated.

- **Four successive authentication intervals are set with location information, and the user pushes a button on each interval.**
  Figure 3 is an example of the setting. In this example, to be authenticated successfully, a user should pass through points A, B, C, and D in sequence, and push a button five times between points A and B, three times between B and C, one time between C and D, and two times between D and E.
- **Four authentication points are set with time information, and the user passes through the points at the appointed speed.**
  Figure 4 is an example of the setting. In this example, to be authenticated successfully, a user should move at 10 kilometers per hour (km/h) after 30 seconds from starting authentication, 20km/h after 60 seconds, 15km/h after 90 seconds, and 25km/h after 120 seconds.

When using authentication points (or intervals) with location information outside, backup secret information should be set in case the user cannot pass through one of authentication points due to road construction.

## 3.4 Informing User of Current Status

If the authentication system uses time or distance information to set authentication points or intervals, it should show time information (elapsed time since starting authentication) or distance information (travel distance from starting authentication) to the user attempting authentication by some means.

When a walking user attempts authentication, it is not very secure to show time or distance information on a mobile device like a PDA. The attacker could speculate about authentication points or intervals by seeing that the user refers to the device around certain places. To enhance security of walking user authentication, time information or distance information is given to a user by a head-mounted display or headphones.

On the contrary, if a user attempting authentication rides a car or a two-wheeled vehicle, it is safe for time or distance information to be shown on a device located by the speed indicator. In this case, the user moves his/her eyes only when referring to the device, and therefore it is not conspicuous. Additionally it is natural for the user to refer to the speed indicator.

## 3.5 Setting of the Margin

The authentication point needs a margin in case a user's action deviates from an authentication point. For example, if an authentication point is set as 30 seconds after starting authentication and a margin is set as 5 seconds, a user is allowed to perform an authentication action during 25 and 35 seconds after starting authentication. The system based on the proposed method has the margin with radius *pMargin* for the location-based authentication point, *tMargin* seconds margin for the time-based authentication point, and *dMargin* long margin for the distance-based authentication point.

The authentication system based on the proposed method should also consider measurement error. When using location information to set authentication points or intervals, the system based on the proposed method needs not only *pMargin* but also the margin *pError* for location measurement error. If the location measurement error were large, the system would recognize that the user who actually passed through an authentication point $point_i$ did not pass through $point_i$ and the user would fail to be authenticated. The system recognizes that a user passed through $point_i$ if he/she passed through $point_i$ with *pMargin*+*pError* radius.

Although distance information also has measurement error, distance information does not require error margin. Users perform authentication actions, referring to distance information given by the system, which includes measurement error. For example, when actual travel distance is 90 meters and measured travel distance is 100 meters, the user is informed that travel distance is 100 meters and he/she does not know the actual travel distance.

When using mobile speed at an authentication point as an authentication action, the system needs the margin considering the error α and β described below.

- **Absolute value of difference between appointed speed and passing speed (α)**
  Small α shows that a user can control passing speed. If α were large and the margin small, the possibility of retrying the authentication action (authentication retry rate) would be higher.
- **Absolute value of between passing speed and speed that a user recognizes as passing speed (β)**
  Although measured speed is shown to the user, mobile speed varies frequently and therefore the user may misread mobile speed at the authentication point. Small β shows that a user perceives mobile speed precisely. If β were large and the margin small, the possibility of retrying authentication (authentication fault rate) would be higher.

Considering α and β, when the appointed speed is $v_j$ km/h at an authentication point $point_j$, the system recognizes that a user whose passing speed at $point_j$ is $v_j$ km/h±*vMargin* satisfies the appointed speed.

## 3.6 Requirement of Devices

When using location information for authentication, the system based on the proposed method needs a device that can measure the location of a user attempting authentication. When a user attempts authentication outdoors, the client needs a GPS unit that can measure the user's location.

When a user attempts authentication indoors, the authentication system based on the proposed method needs a system that measures the user's location by radio waves from the client [8].

If distance information is used for authentication, required devices are the same as using location information because distance information is able to be obtained by using location information. When a user rides a car or a two-wheeled vehicle, travel distance is also measured with the rotation number of the wheels and in this case the client needs a device that measures the rotation number of wheels. If the system authenticates a walking user with distance information, pedometer measurement can be used instead of travel distance by embedding a pedometer in the client. Time information does not require a specific device because most electronic devices have a timer.

The client needs a mobile communication device to communicate with the server. The client also needs the device to measure the operation status described in section 3.2, and the device described in section 3.4, which gives time or distance information to the user.

## 3.7 Retry of Authentication Actions

When using location information to set authentication intervals or points, a user once moves to out of *interval*$_j$ (or *point*$_j$) and moves to *interval*$_j$ again to retry an authentication action at *interval*$_j$.

If the system based on the proposed method uses time or distance information to set authentication points or intervals, a timer or an odometer is reset at a certain interval to retry authentication. For example, when the system resets elapsed time at 60 seconds, and 20 and 30 seconds from starting authentication are set as authentication points, the system checks a user's action at 20 seconds until it corresponds with the correct authentication action. After it corresponds with the correct authentication action, the system checks user's action at 30 seconds in the same way.

When authentication takes a long time, usability is enhanced instead of lowering security slightly by telling a user whether authentication actions performed are correct or incorrect on the way to the destination. The system tells a user whether performed authentication actions are correct or incorrect after a certain period of time (or distance), but not tell after passing through an authentication point (or interval) to avoid that an attacker gets a clue of the authentication point (or interval).

# 4 EXPERIMENT

## 4.1 Overview

We experimented to confirm the availability of our authentication method. In chapter 3, we explained two kinds of authentication action: (1) authentication action based on mobile speed, (2) authentication action based on button status. We measured two metrics described below when the system based on the proposed method adopts these actions:

(a) authentication retry rate
(b) authentication failure rate.

To minimize the influence of measurement errors, we chose a time-based authentication point since its measurement error is less than the others. Experiment tasks are designed not only to measure the authentication retry rate and authentication failure rate, but also to illuminate measurement errors and appropriate margins (see section 3.5). The details of the experiments are described as follows.

## 4.2 Authentication experiment using mobile speed as an authentication action

**Experimental Procedure**

The experiment was conducted in the following manner to examine error α and β, and relationships among *vMargin*, authentication retry rate, and authentication failure rate.

**(Step 1)** An authentication point was set at 30 seconds after starting authentication. Each subject performed a practice run about twice and set the appointed speed arbitrarily.

**(Step 2)** A subject tried to pass the authentication point at the appointed speed.

**(Step 3)** A subject marked the passing speed at the authentication point.

**(Step 4)** Repeated step 2 and step 3 10 times.

We asked 8 subjects to try the experimental authentication by bicycle on a straight road. During the trial, current speed and elapsed time were shown on the PDA fixed on the bicycle. Current speed was measured by GPS unit every second. Note that displaying the actual speed had considerable time lag, and subjects were told about this phenomenon before their trials.

**Results of the experiment**

Boxplots of error α and β are shown in

Figure 5. Error α shows the difference between the appointed speed and the passing speed recorded on PDAs. 14.9% of α are larger than 1.0km/h, and 4.1% of α are larger than 1.5km/h. Error β is the difference between the recorded speed and the speed that was marked by subjects at step 3. 6.8% of β are greater than 0.5km/h, and 1.4% of β are greater than 1.0km/h.

In the case where *vMargin* is set as 1.0km/h, the authentication retry rate is 14.9% because 14.9% of α are greater than 1.0km/h. The authentication failure rate is 2.7%, and the rate is calculated by cases in which subjects erroneously recognize that α is less than 1.0km/h due to error β, although α is actually greater than 1.0km/h. The same as above, when *vMargin* is set as 1.5km/h, the authentication retry rate is 5.0%, and the authentication failure rate is 0%.

The authentication retry rate and authentication failure rate increase *n* times when *n* authentication points are de-

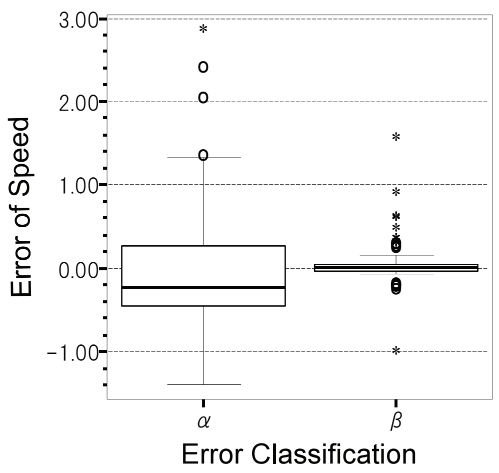

Figure 5: Boxplots of error α and β.

fined. In the case where authentication points are four and *vMargin* is 1.5km/h, the authentication retry rate is 20.0%, and the authentication failure rate is 0%, for instance. Note that the authentication retry rate is the probability of retrying an authentication action at at least one authentication point (or interval), not the probability of retrying authentication from the start. On the contrary, the authentication failure rate is the probability of retrying authentication from the start. The time lag of displaying current speed can be lessened by measuring the current speed with the rotation speed of the wheels, and that may make the authentication retry rate and the authentication failure rate smaller. As a result of the experiment, authentication action based on mobile speed is available.

## 4.3 Authentication experiment using button pushing as an authentication action

**Experimental Procedure**

The experiment was conducted as below to illustrate the relationship between *tMargin* and the authentication failure rate.

**(Step 1)** Each subject set two authentication points by time between 1 second and 30 seconds.

**(Step 2)** Each subject tried to push a button at the authentication points.

**(Step 3)** Each subject marked points where he/she failed to push a button.

**(Step 4)** Each subject repeated steps 2 and 3, 10 times.

We asked 10 subjects to try the experimental authentication by walking on a straight road. Each subject touched a GUI button displayed on a PDA screen, instead of pushing an actual button. During the trials, each subject held a PDA, watched the elapsed time shown on the screen, and pushed the button at authentication points. The PDA recorded the elapsed time when subjects pushed the button.

**Results of the experiment**

When *tMargin* is set as 0 seconds, the authentication retry rate is 7.5%. The rate is the probability that subjects fail to push the button at an authentication point and retry pushing the button at the point. The authentication failure rate is 3.5%. The rate is the probability that subjects erroneously think that they pushed the button at an authentication point, while they actually failed to push the button at the authentication point. The same as above, when *tMargin* is set at 1 second, the authentication retry rate is 4.5%, and the authentication failure rate is 1.0%. None of the subjects pushed the button before they passed authentication points, and therefore there is no need to set *tMargin* before authentication points.

If authentication points are four and *tMargin* is set as 1 second, the authentication retry rate is 18.0%, and the authentication failure rate is 4.0%. Note that the authentication retry rate is the probability of retrying the authentication action at at least one authentication point (or interval), not the probability of retrying authentication from the start. Some subjects mentioned the button on the LCD screen is not suitable for pushing. Using a decent button may make the authentication retry rate and authentication failure rate smaller. The result of this experiment shows that authentication action based on button pushing is available.

# 5 DISCUSSION

The proposed method has a large number of combinations of secret information. The number of combinations of secret information is the "number of authentication action patterns × the number of authentication interval (or point) patterns". For example, if 4 authentication intervals are set and an authentication action is set as pushing a button from 1 to 10 times, the number of authentication action patterns is $10^4 = 10{,}000$ and the number of combinations of secret information is "10,000 × the number of the authentication interval patterns". The number of authentication interval (or point) patterns is increased by extending the distance (or time) from the beginning of authentication to the end. Though in our experiment we set time for authentication as 30 seconds to make the experiment easy, time for authentication should be set as more than 60 seconds to enhance security in actual use.

In our experiment, we did not examine FAR (False Acceptance Rate), which is often used to evaluate biometrics authentication, because the FAR of the proposed method depends on the number of combinations of secret information, which can be increased by changing authentication interval (or point) patterns or authentication action patterns. The FRR (False Rejection Rate) is almost the same as the authentication failure rate.

In the proposed method, the amount of secret information that the user should remember to be authenticated tends to be larger than other authentication methods. The usability of the proposed method is not very high because a user remembers some authentication intervals (or points) and an authentication action to be authenticated. Our method is more conscious of security than usability because it is assumed to be used when high security is needed (i.e. authenticating a user who enters important places such as a secret data storage room, a military installation, or a nuclear power plant.)

## 6 RELATED WORK

There are some authentication methods that use location information or user's actions separately, but there is no research that has proposed an authentication method combining them.

Authentication methods based on user's actions authenticate users who perform actions that correspond with preset actions. An authentication method using a user's hand motion [12] or tracks made by moving a PDA [11] are based on a user's actions. These methods are robust for theft and able to change secret information. But the attacker can observe authentication actions of these methods as opposed to the proposed method. There are some authentication methods of which an attacker can hardly observe the authentication actions. Authentication methods using a user's brain signals [13] or eye gaze to choose pictures containing secret information [3] use invisible authentication actions. However, there methods need installation space for authentication hardware.

There are some authentication methods that use location information to authenticate users. They control access right by a user's current location [2] [10]. For example, if a user is at a place where outsiders cannot enter, an authentication system recognizes the user as an insider and grants various access rights. On the contrary, the system restricts access right of users who are at a place where an outsider can enter. These methods are suitable for access control of a remote system, but not for authentication of entering important places such as a secret data storage rooms. The concept of these methods is applicable to the proposed method with using places where an outsider cannot enter as authentication points, and it enhances the security of the proposed method.

Ishihara et al. [5] proposed an authentication method using a user's location history. In their method, a user is authenticated by answering about places where he/she was. For example, a user indicates places on a map where he/she was 10, 20, 30, and 40 minutes ago to be authenticated. However, this method is vulnerable to tailing. This method uses location information as secret information, while the proposed method not only uses location information as secret information but also uses it when a user inputs secret information.

## 7 CONCLUSIONS

This paper proposed an authentication method combining spatiotemporal information and actions and explained the authentication system based on the proposed method. In the proposed method, a user is authenticated when he/she performed actions corresponding with correct authentication actions on authentication intervals (or points). Our future work is to evaluate the usability and security of the proposed method changing spatiotemporal information and actions variously. Moreover, to evaluate difficulty of identifying authentication actions, we will examine probability that an attacker is authenticated successfully when the attacker observes users' authentication actions by an experiment.

## REFERENCES

[1] J. Daugman, High Confidence Visual Recognition of Persons by a Test of Statistical Independence, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.15, No.11, pp.1148-1161 (1993).

[2] D. Denning, and P. MacDoran, Location-based Authentication: Grounding Cyberspace for Better Security, In Internet besieged: countering cyberspace scofflaws, pp.167-174, ACM Press/Addison-Wesley Publishing Co. (1997).

[3] B. Hoanca, and K. Mock, Secure Graphical Password System for High Traffic Public Areas; Proc. Eye Tracking Research and Applications Symposium, pp.27-29, San Diego, California (2006).

[4] M. S. Hwang, and L. H. Li, A New Remote User Authentication Scheme Using Smart Cards, IEEE Transaction on Consumer Electronics, Vol.46, No.1, pp.28–30 (2000).

[5] Y. Ishihara, and H. Koike, Path-Pass: The Authentication System Using Location Information, Computer Security Symposium (CSS2006), Kyoto, Japan (2006) (in Japanese).

[6] A. Jain, L. Hong, and R. Bolle, On-Line Fingerprint Verification, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.19, No.4, pp.302-314 (1997).

[7] M. Just, and P. Oorschot, Addressing the Problem of Undetected Signature Key Compromise, Proc. Network and Distributed System Security Symposium, San Diego, California (1999).

[8] T. Kitasuka, T. Nakanishi, and Akira Fukuda, Wireless LAN based Indoor Positioning System WiPS and Its Simulation, IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'03), pp. 272-275, Victoria, Canada (2003).

[9] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino, Impact of Artificial “Gummy” Fingers on Fingerprint Systems, Optical Security and Counterfeit Deterrence Techniques IV, Proceedings of SPIE Vol.4677, pp.275-289 (2002).

[10] J. Noda, M. Takahashi, I. Hosomi, H. Mouri, Y. Takata, and H. Seki, Integrating Presence Inference into Trust Management for Ubiquitous Systems, Proc. 11th ACM Symposium on Access Control Models and Technologies (SAC-MAT2006), pp.59-68, Lake Tahoe, California (2006).

[11] M. Ohta, E. Namikata, S. Ishihara, and T. Mizuno, Individual Authentication for Portable Devices using Motion Features, Proc. the 1st International Conference on Mobile computing and Ubiquitous networking (ICMU2004), pp.100-105, Yokosuka, Japan (2004).

[12] R. Osada, S. Ozaki, T. Aoki, and H. Yasuda, A Real Time Personal Verification System Based on Individual Characteristic Extraction of Hand Motion, The Transactions of the Institute of Electronics, Information and Communication Engineers, D-II, J84-D-II(2), pp.258-265 (2001) (in Japanese).

[13] J. Thorpe, P. Oorschot, and A. Somayaji, Passthoughts: Authenticating with Our Minds, Proc. the 2005 workshop on new security paradigms, pp.45-56, Lake Arrowhead, California (2005).

[14] A. Weaver, Biometric Authentication, IEEE Computer, Vol.39, No.2, pp.96-97 (2006).