An Authentication Method with Spatiotemporal Interval and Partial Matching

Masateru Tsunoda
Toyo University
Saitama, Japan
tsunoda@ieee.org

Kyohei Fushida
Nara Institute of Science and Technology
Nara, Japan

Yasutaka Kamei
Kyushu University
Fukuoka, Japan
kamei@ait.kyushu-u.ac.jp

Masahide Nakamura
Kobe University
Hyogo, Japan
masa-n@cs.kobe-u.ac.jp

Kohei Mitsui, Keita Goto, Kenichi Matsumoto
Nara Institute of Science and Technology
Nara, Japan
matumoto@is.naist.jp

***Abstract*—In past research, we proposed an authentication method that combines actions with spatiotemporal information such as location, time, and distance. With the method, a user succeeds in authentication when he/she performs preset actions such as pushing button *n* times on preset intervals defined by spatiotemporal information. In this paper, we improve the authentication method using a partial matching method. We propose two kinds of partial matching methods for pushing button and interval. A type I method assumes the number of pushing button is sometimes less than preset count, but the number never exceeds it, and a user never pushes the button out of preset areas. A type II method assumes the number of pushing button is less or more than preset count occasionally, and a user pushes the button out of preset areas. We showed how to calculate FAR when the type I or II is applied. In the experiment, we compared the type I and II methods with a conventional method to evaluate their security. As a result, the type I method improved false acceptance rate (FAR) from 0.097% to 0.053%. The type II method improved FAR from 0.097% to 0.035%.**

Keywords—spatiotemporal information; spatiotemporal character; partial match authentication; authentication interval; ubiquitous computing

I. INTRODUCTION

Recently, smart card authentication and biometrics authentication are widely used as high secure authentication. In smart card authentication [10], users are authenticated with a smart card which 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 [6]. Biometrics authentication uses the fingerprint [15] or the iris [1] to authenticate users. Biometrics authentication is safe because it is 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.

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 combines an authentication method based on "What you know" with "What you have." Smart card authentication and biometrics authentication are authentication methods based on "What you have." Entering a password with a keyboard is a common method based on "What you know." However, entering password is not very secure because a password can be discovered by others through shoulder surfing.

To enhance security of authentication, we proposed an authentication method that combines actions with spatiotemporal information such as location, time, and distance [17][18]. With the method, a user performs specific actions on several points that are defined by location, time, or distance. Figure 1 shows an example of the authentication combining pushing button and location information. If a user pushes a button of a device on points A and B, C, and D (authentication point), the authentication succeeds and he/she can enter the data center. 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.

To enhance usability of the method, we proposed the authentication method using the authentication interval [17]. The authentication interval is an alternative element to the authentication point. When using the interval, a user pushes the button five times between points A and B (interval I), three times between points B and C (interval J), and one time between points C and D (interval K), to be authenticated, as shown in Figure 2. Using the interval does not require severe timing of performing actions, and therefore the usability is higher than using the point.

However, it is not clear whether using the interval is higher security than using the point or not. In case that using the interval is not higher security than the point, we improve

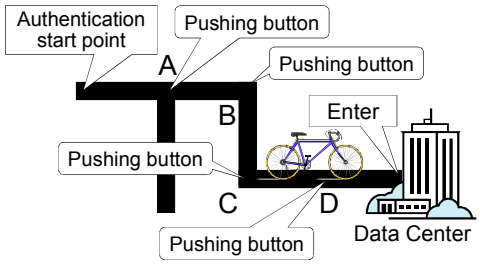

Figure 1. Authentication method with authentication point and action

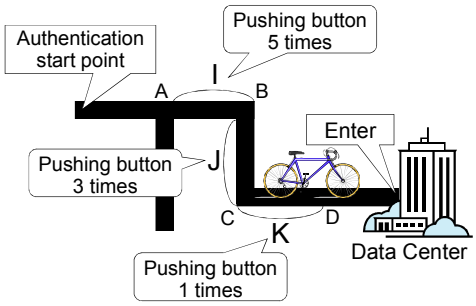

Figure 2. Authentication method with authentication interval and action

authentication method using the interval. In the experiment, to make it easy, we do not conduct the experiment with actual users, but assume false rejection rate (FRR) of the improved method is same as using the point. Also, we show what is required for the improved method to make the FRR same as using the point. So, we set the following research questions:

- **RQ1**: When assuming false rejection rate (FRR) is same, which has higher security, using authentication interval, or using authentication point?
- **RQ2**: When assuming FRR is same, which has higher security, the improved authentication method, or using authentication point?
- **RQ3**: What is required for the improved method to make the FRR same as using the point?

II. SPATIOTEMPORAL BASED AUTHENTICATION

A. Definitions

1) Authentication Area: We defined authentication area as shown in Figure 3. Definitions of each point and interval are as follows:

Location point/interval: When s is set as arbitrary location, authentication area determined by s is called location point. When s_s and s_e are set as arbitrary locations, authentication area determined by the interval between s_s and s_e is called location interval.

Time point/interval: When s is set as arbitrary elapsed time from authentication start point, authentication area determined by s is called time point. For example, "Time point A is 15 seconds" means the point when s is15 seconds. When s_s and s_e are set as arbitrary elapsed time from authentication start point, authentication area determined by the interval between s_s and s_e is called time interval. For example, "Time interval A is from 15 seconds to 20 seconds" means the interval between the point when s_s is 15 seconds and the point when s_e is 20 seconds.

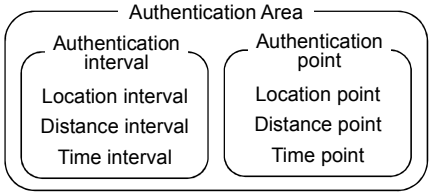

Figure 3. Inclusive relation of authentication area

Distance point/interval: When s is set as arbitrary travel distance from authentication start point, authentication area determined by s is called distance point. For example, "Distance point A is 100 meters" means the point where s is 100 meters. When s_s and s_e are set as arbitrary travel distance from authentication start point, authentication area determined by the interval between s_s and s_e is called distance interval. For example, "Distance interval A is from 100 meters to 150 meters" means the interval between the point where s_s is 100 meters and the point where s_e is 150 meters.

2) Authentication Action: Authentication action is an action which a user should perform to be authenticated. When authentication point is used, candidates of the actions are performing and not performing an action. For example, pushing a button and not pushing a button are the candidates. In contrast, when authentication interval is used, the candidates are performing an action n times. For example, pushing a button two times and pushing a button four times are the candidates.

3) Spatiotemporal Character and String: Spatiotemporal character is a set of an authentication area p and an authentication action a $\langle p, a \rangle$. Spatiotemporal string is sequence of k spatiotemporal characters. When p_i is an authentication area and a_i is an authentication action, a spatiotemporal string is denoted as $\langle p_1, a_1 \rangle ... \langle p_i, a_i \rangle ... \langle p_k, a_k \rangle$. For example, the spatiotemporal string in Figure 1 is ⟨A, "pushing button"⟩ ⟨B, "pushing button"⟩ ⟨C, "pushing button"⟩ ⟨D, "pushing button"⟩, and the spatiotemporal string in Figure 2 is ⟨I, "pushing button five times"⟩ ⟨J, "pushing button three times"⟩ ⟨K, "pushing button one time"⟩.

Preset action is an authentication action which is prescribed on an authentication system, and user action is an authentication action which a user actually performs. Likewise, we classify authentication area into preset/user area, spatiotemporal character into preset/user character, and spatiotemporal string into preset/user string. Spatiotemporal based authentication does not care sequence of characters in a user string, because it does not enhance security of authentication very much.

B. Partial Matching Method

A partial matching method judges authentication as successful when the number of mismatch characters is smaller than **allowed mismatch number** d between a preset string and a user string. The mismatch character means when a preset

TABLE I. AUTHENTICATION RESULTS WHEN THE PARTIAL MATCHING METHOD IS USED

User string	Result	Reason
⟨K, 1⟩ ⟨I, 5⟩ ⟨J, 3⟩	Success	Sequence is not cared
⟨I, 5⟩ **⟨J, 5⟩** ⟨K, 1⟩	Success	One mismatch character
⟨I, 5⟩ ⟨J, 3⟩ **⟨L, 1⟩**	Success	One mismatch character
⟨I, 5⟩ ⟨J, 3⟩	Success	One mismatch character
⟨I, 5⟩	Failure	Two mismatch character
⟨I, 5⟩ ⟨J, 3⟩ ⟨K, 2⟩ **⟨L, 1⟩**	Failure	Too long string

TABLE II. AUTHENTICATION RESULTS WHEN THE TYPE I METHOD IS USED

User string	Result	Reason
⟨I, 5⟩ **⟨J, 2⟩** ⟨K, 1⟩	Success	One mismatch character
⟨I, 5⟩ ⟨J, 3⟩ **⟨K, 2⟩**	Failure	Too many pushing button
⟨I, 5⟩ **⟨J, 1⟩** ⟨K, 1⟩	Failure	Too less pushing button
⟨I, 5⟩ ⟨J, 3⟩ **⟨L, 1⟩**	Failure	Out of preset area

TABLE III. AUTHENTICATION RESULTS WHEN THE TYPE II METHOD IS USED

User string	Result	Reason
⟨I, 5⟩ **⟨J, 2⟩** ⟨K, 1⟩	Success	One mismatch character
⟨I, 5⟩ ⟨J, 3⟩ **⟨K, 2⟩**	Success	One mismatch character
⟨I, 5⟩ ⟨J, 3⟩ **⟨K, 3⟩**	Failure	Too many pushing button
⟨I, 5⟩ **⟨J, 1⟩** ⟨K, 1⟩	Failure	Too less pushing button
⟨I, 5⟩ ⟨J, 3⟩ **⟨L, 1⟩**	Success	One mismatch character
⟨I, 5⟩ ⟨J, 3⟩ **⟨L, 2⟩**	Failure	More than 1 times out of preset area

character is denoted as $\langle p, a \rangle$, and a user character is denoted as $\langle q, b \rangle$, the preset action a does not correspond to the user action b, or the preset area p does not correspond to the user area q. In the following, we call the method as the **normal matching method**.

The normal matching method allows that a user string is shorter than a preset string. However, it does not allow that a user string is longer than a preset string, because inputting many characters enhances probability of matching a preset string and a user string. Table I shows examples of user strings and their results, when authentication interval is used, $d = 1$, and a preset string is ⟨I, 5 (pushing button 5 times)⟩ ⟨J, 3⟩ ⟨K, 1⟩.

C. Partial Matching Method for Pushing Button and Interval

As stated in section I, it is not clear whether using the interval is higher security than using the point or not. In case that using the interval is not higher security than the point, we enhance security of the normal matching method. That is, we propose two kinds of partial matching methods specialized for pushing button and interval.

A **type I** method classifies behavior of a user as followings. In the followings, the preset action is "pushing button n times."

- Case I-1: Pushing button n - 1 times.
- Case I-2: Pushing button out of preset areas.
- Case I-3: Pushing button less than n - 1 times.
- Case I-4: Pushing button more than n times.

The method assumes that a user sometimes perform case I-1 erroneously, but never does other cases. So, the method admits case I-1 as a mismatch character, but does not admit other cases. That is, if a user string includes case I-2, I-3 or I-4, authentication fails regardless allowed mismatch number. Table II shows examples of user strings and their results, when the type I method is used.

A **type II** method classifies behavior of a user as followings. In the followings, the preset action is "pushing button n times."

- Case II-1: Pushing button n - 1 times.
- Case II-2: Pushing button n + 1 times.
- Case II-3: Pushing button 1 times out of preset areas.
- Case II-4: Pushing button more than n + 1 times.
- Case II-5: Pushing button less than n – 1 times.
- Case II-6: Pushing button more than 1 times out of preset areas.

The method assumes that a user sometimes perform case II-1, II-2 and II-3 erroneously, but never does other cases. So, the method admits case II-1, II-2, and II-3 as a mismatch character, but does not admit other cases. Table III shows examples of user strings and their results, when the type II method is used.

III. SPATIOTEMPORAL BASED AUTHENTICATION SYSTEM

Spatiotemporal based authentication system consists of a device which is used to input user characters, and a server which authenticates the user. Each user has user ID (identifier), and a preset string is set for each ID. To enhance security of authentication, tamper resistant devices should be used. Authentication procedure is as follows:

1. The user orders the device to start authentication on arbitrary point.
2. The user inputs user characters to the device, travelling to the destination.
3. When the user arrives at the destination, he/she orders the device to finish authentication.
4. When the device receives the finish command, the device sends the user string and user ID to the server.
5. The server compares user sting and preset string to authenticate the user, and sends the result to the device.

When authentication fails t times successively, the user account is locked. we call t **permitted trial count**. Liken to password, how many times a user tries to input passwords.

When time or distance area is used, a device which informs elapsed time or travel distance to a user is needed. When location area is used and authentication is performed outdoors, GPS (Global Positioning System) is needed (Using Quasi-Zenith Satellite System [6] enhances the measurement accuracy). When authentication is performed indoors, locating system such as UWB (Ultra Wide Band) based one [4] is needed. Travel distance is calculated by location information.

IV. SECURITY EVALUATION

This section explains formalized security evaluation of spatiotemporal based authentication. **Attacker** is a person who tries authentication improperly, although he/she is not permitted to be authenticated. To make deriving equations easy, we assumed an attacker has the device for authentication, and he/she knows the followings:

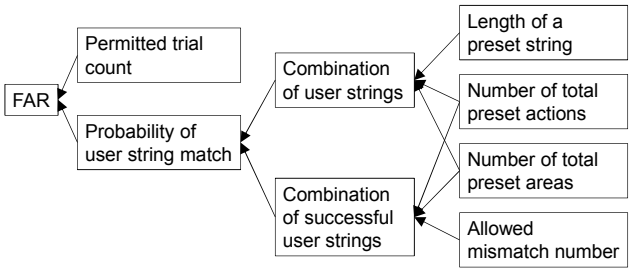

Figure 4. Dependency of FAR

- User ID, length of the preset string, and candidates of preset areas and actions.
- A preset string includes either time area, distance area, or location area.
- A preset string does not include duplicative preset area.

Note that the assumptions are disadvantaged for evaluating security of spatiotemporal based authentication.

A. Probability of Successful Authentication for Attackers

1) False Acceptance Rate (FAR): False acceptance rate (FAR) is probability that an attacker succeeds in authentication, when t is permitted trial count, and the attacker inputs user strings t times. FAR is calculated by:

$$\mathrm{FAR} = 1 - \prod_{i=1}^{t} (1 - \alpha_i) \tag{1}$$

In the equation, α_i is **probability of user string match** on *i-th* authentication trial by the attacker. α_i is probability that the attacker succeeds in authentication when he/she inputs a user string randomly. So, $(1 - \alpha_i)$ is probability that the attacker fail to be authenticated on *i-th* trial, and $\prod_{i=1}^{t}(1-\alpha_i)$ is probability that the attacker fail to be authenticated *t* times in a row. FAR is calculated as complementary event of it.

α_i is calculated by:

$$\alpha_i = \frac{v}{w - i + 1} \tag{2}$$

In the equation, *w* is **combination of user strings** which an attacker makes, and *v* is **combination of successful user strings**. On the denominator, *i* - 1 is subtracted from *w*, because on *i-th* trial, the attacker does not use the user string which is rejected before the trial.

To illuminate how to derive *w* and *v*, we first use examples of user strings, and then we show generalized equations. In the examples, we assumed the followings:

- The preset string is ⟨A5 (pushing button 5 time)⟩ ⟨B5⟩ ⟨C5⟩ ⟨D5⟩.
- Candidates of preset areas: A to Z (26 types).
- Candidates of preset actions: 1 to 9 (9 types).

First, we explain how to derive *w* and *v*, assuming length of a user and preset string is same. Then, we explain that, assuming a user string is shorter than a preset string. Deriving *w* when applying the normal matching method is explained in [18].

TABLE IV. COMBINATION NUMBER OF SUCCESSFUL USER STRINGS

d	*g*	*g - h*	Case 1	Case 2	Case 3
0	0	0	${}_4C_4$	${}_4C_0 \times 2^0$	${}_{22}C_0$
1	1	0	${}_4C_3$	${}_3C_0 \times 2^0$	${}_{22}C_1$
	0	1	${}_4C_4$	${}_4C_1 \times 2^1$	${}_{22}C_0$
2	2	0	${}_4C_2$	${}_2C_0 \times 2^0$	${}_{22}C_2$
	1	1	${}_4C_3$	${}_3C_1 \times 2^1$	${}_{22}C_1$
	0	2	${}_4C_4$	${}_4C_2 \times 2^2$	${}_{22}C_0$
3	3	0	${}_4C_1$	${}_1C_0 \times 2^0$	${}_{22}C_3$
	2	1	${}_4C_2$	${}_2C_1 \times 2^1$	${}_{22}C_2$
	1	2	${}_4C_3$	${}_3C_2 \times 2^2$	${}_{22}C_1$
	0	3	${}_4C_4$	${}_4C_3 \times 2^3$	${}_{22}C_0$

Figure 4 illustrates dependency of FAR. Each term are explained in the followings.

2) Posssible User Strings: ⟨J5⟩ ⟨K5⟩ ⟨L5⟩ ⟨M5⟩ is one of user strings. Combination number of user area included in user strings (made by the attacker) is ${}_{26}C_4$ (extracting four characters from 26 characters). For each case, one out of nine user actions is performed on the four user area. So, combination of user strings is ${}_{26}C_4 \times 9^4$. Generalized equation of combination of user strings w_f is calculated by:

$$w_f = {}_r C_m \, c^m \tag{3}$$

In the equation, *m* is length of a preset string, *r* is the number of total preset areas, and *c* is the number of total preset actions.

3) Successful User Strings When Type I Method is Applied: This part explains how to derive combination of successful user strings, when the type I method is applied. When allowed mismatch number *d* is zero, the combination number is one. When *d* is one, successful user strings may include one mismatch character (e.g., ⟨[A-D]4⟩ by regular expressions) out of four. So, the combination number is $1 + {}_4C_1$ (sum of cases that *d* is zero and one). Therefore, when *m* is length of a preset string, combination of successful user strings v_f is calculated by:

$$v_f = \sum_{h=0}^{d} {}_m C_h \tag{4}$$

Note that we do not care the case that the preset action is "pushing button one (minimum) times," to make the derivation of the equation easy. This is disadvantaged for evaluating security of spatiotemporal based authentication.

4) Successful User Strings When Type II Method is Applied: This part explains how to derive combination of successful user strings, when the type II method is applied. Examples in the explanation assumed the followings:

- A0: Allowed mismatch number *d* is two, and actual number of mismatch characters is also two.
- A1: One of mismatch character is that user area is match, but user action is mismatch (e.g., ⟨[A-D][46]⟩ by regular expressions).
- A2: The other mismatch character is that user area is mismatch (e.g., ⟨[E-Z]1⟩).

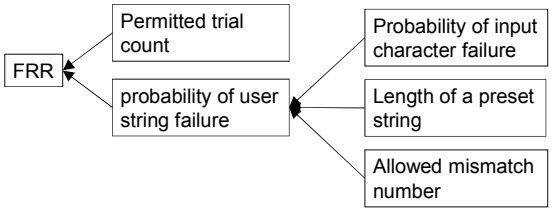

Figure 5. Dependency of FRR

Case 1 - From A1, three user areas are match in user strings. So, the combination number is ${}_4C_3$ (extracting three out of four characters from the preset string).

Case 2 - From A2, one out of three characters of case 1 is the action mismatch character, and therefore its combination number is ${}_3C_1$. Action mismatch means when a preset action is "pushing button n times," user actions are "pushing button n - 1 times" or "pushing button n + 1 times" (The two actions are allowed). So, the combination number is ${}_3C_1 \times 2^1$.

Case 3 - Combination of area mismatch characters: One area is selected from areas [E-Z] (22 areas except for [A-D]), and allowed user action is only "pushing button one time." So, the combination number is ${}_{22}C_1$.

Case 1, 2 and 3 are independent events. So, the combination of successful user strings is ${}_4C_3 \times {}_3C_1 \times 2^1 \times {}_{22}C_1$ (multiplying combination numbers of them).

Let g be the number of A0 characters, and let h be the number of A1 characters (the number of A2 characters is $g - h$). Based on Table IV, when m is length of a preset string, d is allowed mismatch number, and r is the number of total preset areas, combination of case 1 is ${}_mC_{n-g+h}$, that of case 2 is ${}_{m-g+h}C_h \times 2^h$, and that of case 3 is ${}_{r-m}C_{g-h}$. Combination number of each row in Table IV is calculated by multiplying combination numbers of case 1, 2, and 3. Rows in Table IV are independent events. So, combination of successful user strings v_f is sum of combination numbers of the rows. It is calculated by:

$$v_f = \sum_{g=0}^{d} \sum_{h=0}^{g} {}_m\mathrm{C}_{m-g+h}\, {}_{m-g+h}\mathrm{C}_h\, 2^h\, {}_{r-m}\mathrm{C}_{g-h} \tag{5}$$

Note that we do not care the cases that the preset action is "pushing button one (minimum) times," or "pushing button nine (maximum) times," to make the derivation of the equation easy. This is disadvantaged for evaluating security of spatiotemporal based authentication.

5) Short User String: This part explains how to derive combination of user strings w and combination of successful user strings v, when a user string is shorter than a preset string. First, derivation of w is explained. For example, when length of a preset string m is four, and allowed mismatch number d is one, user strings which have three or four characters are successful, if there is no mismatch characters. Let y_i be combination number calculated by equation 3 (In the equation, y_i is w_f) when m is i. The combination of the user strings w is y_3+y_4. So, w is calculated by (Note that y_0 is zero):

$$w = \sum_{i=n-d}^{m} y_i \tag{6}$$

Next, combination of successful user strings v is explained. For example, if length of a preset string m is four, and d is one, authentication succeeds when a user string includes three characters and the number of mismatch characters is zero, or when a user string includes four characters and the number of mismatch characters is less than or equal to one. Let $z_{i,j}$ be combination number calculated by equation 4 or 5 (In the equations, $z_{i,j}$ is v_f) when m is i and d *is* j. The combination of successful user strings v is $z_{3,0}+z_{4,1}$. So, v is calculated by:

$$v = \sum_{i=n-d}^{n} z_{i,i-n+d} \tag{7}$$

Note that the equation does not change whether the type I or II method is applied or not.

FAR is different when an attacker inputs only user strings whose length is same as a preset string (i.e., equation 3, 4, and 5 is adopted), or when he/she inputs the strings whose length is same as or shorter than the preset string (i.e., equation 6 and 7 is adopted). In the experiment, we changed parameters such as the number of authentication area, and adopted higher FAR. Note that this is a disadvantage for evaluation of the spatiotemporal based authentication.

B. Probability of Failure Authentication for Users

In this subsection, failure authentication for users is explained. We assumed that a user knows the preset string linked to his/her user ID.

1) False Rejection Rate (FRR): False rejection rate (FRR) is probability that a user does not succeeds in authentication within t times trials (t is permitted trial count). FRR is calculated by:

$$\mathrm{FRR} = \beta^t \tag{8}$$

In the equation, β is **probability of user string failure**. It is probability that a user does not succeed in authentication due to failing to input a user string on *i-th* authentication trial. Liken to password, it is probability that a user does not succeed in authentication due to mistyping. β is calculated using **probability of input character failure**. Figure 5 illustrates dependency of FRR. Each term are explained in the followings.

2) Probability of Input Character Failure: Probability of input character failure γ is the probability that a user fails to input a user character. Liken to password, it is probability that a user makes mistyping one character. Although γ is generally speculated based on experimental results, this paper set assumed value to γ to make the experiment easy.

Note that when the type I method is applied, the failure means case I-1 (see section II.C) occurs. The failure does not include case I-2, I-3, and I-4, because the method assumes the cases never occur. When the type II method is applied, the failure means case II-1, II-2, or II-3 occurs. The failure does not include case II-4, II-5, and II-6.

3) Probability of User String Failure: Let γ be probability of input character failure, let m be length of a preset string, and let d allowed mismatch number. Probability of user string failure β is probability that a user fails to input more than or

equal to d+1 characters out of m characters in the user string. So, the probability is calculated by:

$$\beta = \sum_{i=d+1}^{m} {}_m\mathrm{C}_i \gamma^i (1-\gamma)^{m-1} \quad (9)$$

In the equation, ${}_m\mathrm{C}_i\ \gamma^i\ (1 - \gamma)^{m-1}$ is probability that a user fails to input i out of m characters, and it follows a binomial distribution. β is sum of the probability when a user fails to input more than or equal to d+1 characters.

C. Number of Candidates of Preset Areas

When authentication interval is applied, the number of total preset areas r is calculated by:

$$r = \left\lfloor \frac{s}{u} \right\rfloor \quad (10)$$

In the equation, s is total elapsed time or total travel distance which an authentication system assumes, and u is unit size of an authenticating interval. The number of candidates for the case of applying authentication point is explained in [18].

V. EXPERIMENT

A. Overview

To answer RQ1 to RQ3, we evaluated security of spatiotemporal based authentication. When security of an authentication method is evaluated, both FAR and FRR should be considered because there is trade-off between FAR and FRR. However, we did not evaluate FRR of the type I and II method, to make the experiment easy. We evaluated security of the type I and II method, assuming the FRR is same as conventional method (authentication method with authentication point) (to answer RQ1 and RQ2). Instead of evaluating the FRR, we derived required probability of input character failure, to make security of the type I and II method higher than the conventional method (to answer RQ3).

The experiment was performed by the following steps:

1. Set baselines by evaluating the conventional method (authentication method with authentication point).
2. Compare the normal matching method, the type I method, and the type II method with the baselines, to answer RQ1 and RQ2.
3. Derive required probability of input character failure, to answer RQ3.

We used time area as authentication area. Total elapsed time was set as 60 seconds, and permitted trial count was set as two (The conditions are same as [18]). Unit size of an authentication interval was set as 10 seconds. So, the number of total preset areas was six when authentication interval is used (from equation 10). The number of total preset areas was 59 when authentication point (conventional method) is used (from [18]). Authentication action was "pushing button" when authentication point is used, and the actions were "pushing button n times (n is from 1 to 9)" when authentication interval is used. Length of preset strings was set as four and five, considering FAR and usability.

B. Setting Baselines

To set baselines, we focused on the conventional method (authentication method with authentication point). The baselines mean the cases whose FAR and FRR are sufficient in actual use. Table V shows relationships of FAR, FRR, length of preset strings m, allowed mismatch number d, and probability of user string failure β (from [18]). Considering FAR (security) and β (usability), we set the case "m is four and d is one" and the case "m is five and d is one" as baselines (indicated by boldface). Note that more authentication areas make usability lower because a user should remember more areas. So, the case "m is five and d is one" is preferable to be used when security is severe.

C. Comparison of FAR

We compared the normal matching method, the type I method, and the type II method with the baselines, and identified cases which are more secure than the baselines. "More secure" means FAR is enough smaller and FRR is not larger than the baselines. Table VI, VII, and VIII shows relationships of FAR, FRR, length of preset strings m, allowed mismatch number d, and probability of user string failure β (FRR and β is same as Table V).

Comparing Table V with Table VI, in the normal matching method, there were no case in which FAR was enough smaller and FRR was not larger than the baselines. So, the method is not more secure than the authentication point method, if FRR is same as the conventional method, and the parameters such as d are set as the experiment. Therefore, the answer of RQ1 is "Using authentication point (conventional method) has higher security than using authentication interval."

Comparing Table V with Table VII and VIII, the following cases were more secure than the conventional method (authentication method with authentication point), if FRR is same as the conventional method. So the answer of RQ2 is "The improved authentication method (the type I and type II method) has higher security than using authentication point (conventional method)."

- In the type I method, when m is four, the cases "d is from one to four" had enough smaller FAR than the baseline. We focused on the case "d is four" (indicated by boldface), because it had smallest FRR. In this case, FAR was improved from 0.097% (the baseline) to 0.053%.
- In the type I method, when m is five, the case "d is one" had enough smaller FAR (indicated by boldface) than the baseline. In this case, FAR was improved from 0.011% (the baseline) to 0.003%.
- In the type II method, when m is four, the case "d is one" had enough smaller FAR than the baseline and same FRR as the baseline. (indicated by boldface). In this case, FAR was improved from 0.097% (the baseline) to 0.035%.

D. Required Probability of Input Character Failure

In the case "length of preset strings m is four and allowed mismatch number d is four" of the type I method, FRR is zero, even if probability of input character failure γ is 100%. This is because when m is same as d, authentication succeeds, if all

TABLE V. FAR, FRR, AND PROBABILITY OF USER STRING FAILURE IN THE CONVENTIONAL METHOD [18]

m	4			5		
d	β (%)	FRR (%)	FAR (%)	β (%)	FRR (%)	FAR (%)
0	26.791	7.177	0.000	32.281	10.421	0.000
1	**3.047**	**0.093**	**0.097**	**4.828**	**0.233**	**0.011**
2	0.159	0.000	3.972	0.376	0.001	0.582
3	0.003	0.000	43.845	0.015	0.000	10.216
4	0.000	0.000	100.000	0.000	0.000	60.096
5				0.000	0.000	100.000

TABLE VI. FAR, FRR, AND PROBABILITY OF USER STRING FAILURE IN THE NORMAL MATCHING METHOD

m	4			5		
d	β (%)	FRR (%)	FAR (%)	β (%)	FRR (%)	FAR (%)
0	26.791	7.177	0.002	32.281	10.421	0.001
1	3.047	0.093	0.213	4.828	0.233	0.049
2	0.159	0.000	5.418	0.376	0.001	1.219
3	0.003	0.000	46.382	0.015	0.000	13.387
4	0.000	0.000	100.000	0.000	0.000	62.456
5				0.000	0.000	100.000

TABLE VII. FAR, FRR, AND PROBABILITY OF USER STRING FAILURE IN THE TYPE I METHOD

m	4			5		
d	β (%)	FRR (%)	FAR (%)	β (%)	FRR (%)	FAR (%)
0	26.791	7.177	0.002	32.281	10.421	0.001
1	3.047	0.093	0.011	**4.828**	**0.233**	**0.003**
2	0.159	0.000	0.028	0.376	0.001	0.009
3	0.003	0.000	0.046	0.015	0.000	0.018
4	**0.000**	**0.000**	**0.053**	0.000	0.000	0.024
5				0.000	0.000	0.026

TABLE VIII. FAR, FRR, AND PROBABILITY OF USER STRING FAILURE IN THE TYPE II METHOD

m	4			5		
d	β (%)	FRR (%)	FAR (%)	β (%)	FRR (%)	FAR (%)
0	26.791	7.177	0.002	32.281	10.421	0.001
1	**3.047**	**0.093**	**0.035**	4.828	0.233	0.009
2	0.159	0.000	0.196	0.376	0.001	0.054
3	0.003	0.000	0.584	0.015	0.000	0.174
4	0.000	0.000	0.932	0.000	0.000	0.371
5				0.000	0.000	0.504

user characters are mismatch. So, required γ is equal to or smaller than 100%. The answer of RQ3 for the case is "The requirement for the improved method (the type I method) is to set input character failure as equal to smaller than 100%." Note that the failure does not include case I-2, I-3, and I-4, as stated in section IV. B. 2.

In other cases identified in the previous subsection, their *d* were same as the baseline. From equation 9, when *m* and *d* are same, γ should be same, to make probability of user string failure β (i.e., FRR. See equation 8) same. From [18], γ of the baseline was 7.5%. So, required γ is equal to or smaller than 7.5%. Therefore, the answer of RQ3 for these cases is "The requirement for the improved method (the type I and type II method) is to set input character failure as equal to smaller than 7.5%." Note that the failure does not include case II-4, II-5, and II-6, as stated in section IV. B. 2.

VI. DISCUSSION

In fingerprint authentication, which is widely used as biometrics authentication, false acceptance rate (FAR) is 0.01%, and false rejection rate (FRR) is 0.1% [11]. Assumptions of calculating FAR and FRR are different between fingerprint authentication and spatiotemporal based authentication. Therefore it is not proper that their FAR and FRR are compared straightforwardly, and consider which is more secure method. In addition, FAR and FRR of spatiotemporal based authentication were calculated based on many assumptions, and FAR and FRR may get worse to some extent in actual use. However, FAR and FRR were fairly low in the experiment, and therefore we think spatiotemporal based authentication with the type I or II method is enough secure, if the limitations are considered. Note that FRR was calculated assuming users remember their preset string correctly. If their memories are wrong, FRR gets worse.

An attacker can steal passwords of many users easily, if he/she sets a hidden camera on the authentication place. In contrast, if an attacker tries to steal preset strings of spatiotemporal based authentication, he/she should tailing users many times, and it makes the steal difficult. When password is used for authentication which needs high security, the place of inputting password may be secure, and it makes probability of shoulder surfing low. If spatiotemporal based authentication is used in the same situation, security of spatiotemporal based authentication is enhanced. So, advantage of spatiotemporal based authentication over password does not change.

Spatiotemporal based authentication may be interfered with by an attacker. But an attacker can interfere with most of authentication methods, if he/she attacks users physically. So, it is not drawback of spatiotemporal based authentication. Additionally, an attacker can destroy authentication devices such as fingerprint scanner. These discussions are separate from security of authentication methods.

VII. RELATED WORK

Authentication methods which are robust against shoulder surfing are proposed. For example, there are an authentication method using motion features of a mobile device [13], a method based on users' finger motion [8], a method based on users' gain [14], and a method based on users' body motion [3]. FAR was about 1 to 5% in [13] and [14], and equal-error rate was 4.2% in [3] (FAR was not mentioned in [8]). Although simple comparison of FAR and FRR should be avoided, spatiotemporal based authentication with the type I and II method is enough secure, compared with the methods. Note that the usability of spatiotemporal based authentication is lower than the methods, and application area of them is basically different.

Ishihara et al. [5] proposed an authentication method using history of user location. A user is authenticated, answering locations where his/her visited. However, the method is vulnerable to getting location history by attacker's GPS logger [16]. In contrast, spatiotemporal based authentication is relatively robust against such attack, because it combines authentication action with spatiotemporal information.

Also, there are some authentication methods which use location information. 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. These methods are suitable for access control of a 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 spatiotemporal based authentication method, using places where an outsider cannot enter as authentication areas, and it enhances the security.

VIII. CONCLUSIONS

This paper improved spatiotemporal based authentication using the partial matching method. In spatiotemporal based authentication, a user succeeds in authentication when he/she performs preset actions such as pushing button *n* times on preset intervals defined by spatiotemporal information. To suppress false acceptance rate (FAR), we proposed two kinds of partial matching methods for pushing button and interval. The type I method assumes the number of pushing button is sometimes less than preset count, but the number never exceeds it, and a user never pushes the button out of preset intervals. The type II method assumes the number of pushing button is less or more than preset count occasionally, and a user pushes the button out of preset intervals. We explained equations for calculating FAR when the type I or II method is applied.

In the experiment, we evaluated security of the type I and II method. When length of preset string was four, the type I method improved FAR from 0.097% to 0.053%. In this case, required probability of input character failure was equal to or smaller than 100%. The type II method improved FAR from 0.097% to 0.035%. When the length is five, the type I method improves FAR from 0.011% to 0.003%. In these cases, required probability of input character failure was equal to or smaller than 7.5%.

One of our future works is conducting experiments with actual users to evaluate whether the assumptions of the type I and II method are satisfied or not (i.e., whether case I-2, I-3, I-4, II-4, II-5, and II-6 are occurred or not). We will also evaluate whether required probability of input character failure is satisfied or not, when the type I and II method is applied. The other future work is that we will conduct experiments with actual users to evaluate practical feasibility from the viewpoint of usability.

ACKNOWLEDGMENT

This research was partially supported by the Japan Ministry of Education, Science, Sports, and Culture [Grant-in-Aid for Scientific Research (C) (No.24500079), Scientific Research (B) (No.23300009)].

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] F. Gomez-Caballero, T. Shinozaki, S. Furui, and K. Shinoda, "Person authentication using 3D human motion," Proc. the 2011 joint ACM workshop on Human gesture and behavior understanding (J-HGBU '11), pp. 35–40, Scottsdale, Arizona, 2011.

[4] Hitachi, Ltd., "Verification of new Sensor Net communication system using 3-nanoW/bps low-power UWB radio transmission," http://www .hitachi.com/rd/portal/pdf/news/crl051014nrde_uwbsensornet.pdf

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

[6] Japan Aerospace Exploration Agency (JAXA), Quasi-Zenith Satellite-1 MICHIBIKI, http://www.jaxa.jp/projects/sat/qzss/index_e.html

[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] H. Manabe, and M, Fukumoto, "AwareLESS authentication: insensible input based authentication," CHI '07 Extended Abstracts on Human Factors in Computing Systems (CHI EA '07), pp.2561–2566, San Jose, California, 2007.

[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] K. Mayes,and K. Markantonakis, Smart Cards, Tokens, Security and Applications, Springer, 2008.

[11] National Institute of Standards and Technology, "FpVTE 2003: Results," Biometric Technology Today, vol. 12, issue 8, pp. 7–9, 2004.

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

[13] M. Ohta, E. Namikata, S. Ishihara, and T. Mizuno, "Individual Authentication for Portable Devices using Motion Features," Proc. International Conference on Mobile computing and Ubiquitous networking (ICMU), pp.100–105, Yokosuka, Japan, 2004.

[14] K. Sugiura, Y. Makihara, and Y. Yagi, "Gait Identification based on Multi-view Observations using Omnidirectional Camera," Proc. Asian Conf. on Computer Vision (ACCV), vol. 1, pp. 452–461, Tokyo, Japan, 2007.

[15] R. Ramotowski, Lee and Gaensslen's Advances in Fingerprint Technology, Third Edition, CRC Press, 2012.

[16] Telespial Systems, Inc., TrackStick, http://www.trackstick.com/

[17] M. Tsunoda, K. Mitsui, K. Fushida, Y. Kamei, M. Nakamura, K. Goto, and Kenichi Matsumoto, "An Authentication Method Combining Spatiotemporal Information and Actions," Proc. International Conference on Mobile Computing and Ubiquitous Networking (ICMU), pp.41–49, Tokyo, Japan, 2008.

[18] M. Tsunoda, K. Fushida, Y. Kamei, M. Nakamura, K. Mitsui, Keita Goto, and Kenichi Matsumoto, "An Authentication Method Based on Spatiotemporal Information and Actions," Journal of Japan Society for Fuzzy Theory and Intelligent Informatics, vol. 23, no. 6, pp. 874–881, 2011 (in Japanese).

[19] A. Weaver, "Biometric Authentication," IEEE Computer, vol.39, no.2, pp.96–97.