Extended PIN Authentication Scheme Allowing Multi-Touch Key Input

Tetsuji Takada
The Univ. of Electro-Communications
1-5-1 Chofugaoka, Chofu
Tokyo, 182-8585, JAPAN
zetaka @ computer.org

Yuki Kokubun
The Univ. of Electro-Communications
1-5-1 Chofugaoka, Chofu
Tokyo, 182-8585, JAPAN
kokubun @ az.inf.uec.ac.jp

ABSTRACT

In this paper, we report our trial to make a better form of personal identification number(PIN) authentication for a mobile device. We think that mobile users should be given a more secure alternative authentication because PIN authentication has well-known flaws. However, proposed alternative schemes change the authentication method drastically and that may discomfort mobile users. Our approach is to just change the input operation of PIN authentication by allowing more than one number at a time using a multi-touch enabled screen. We implemented a web-based prototype system and conducted an informal user study using it. The results of the study indicate that PIN input time, input errors and secret memorability of the proposed scheme were no worse than those of conventional PIN authentication. We also discuss the mathematical security level and other advantages of the scheme.

Categories and Subject Descriptors

H.5.m [**Information Interfaces and Presentation (e.g., HCI)**]: Miscellaneous; K.6.5 [**Security and Protection**]: Authentication

Keywords

User authentication, Personal Identification Number, PIN, Password, Smartphone, Multi-touch, Input scheme, Mobile device

1. INTRODUCTION

Personal identification number(PIN) authentication is a simple and well established user authentication system for mobile devices. A PIN is one number, it is often four digits, making it easy to remember for many people. It also makes it easy to input with a numerical keypad only. Thus, PIN authentication is a good scheme for both users and developers.

MoMM 2013, 2-4 December, 2013, Vienna, Austria

On the other hand, PIN authentication has some security issues. Other schemes have been proposed as alternatives to PINs [3, 4, 5]. These schemes change some of the authentication elements: the secret information, the input operation, or the user interface. However, these changes increase user's operational load, so users are hesitant to start using these new schemes. Developers of devices are also hesitant to adopt a novel authentication scheme because they would have to implement it from scratch.

In this paper, we propose a novel PIN authentication scheme that allows users to input multiple numbers at a time by using a multi-touch function. An important point of our proposal is that we just change the input operation and try to leave as they are as many other elements of PIN authentication as possible. In other words, our aim is to improve security while minimizing any increase in difficulty of PIN authentication. We believe that mobile device users will notice the additional load in proposed scheme reasonable for a better alternative authentication and this will motivate them to start using an another authentication method.

2. PROPOSED AUTHENTICATION SCHEME

Why do we have to enter each digit of a PIN one by one? Recent smart phones have a multi-touch enabled screen, and mobile device users have become familiar with touch operation with multiple fingers. Most of us can physically press up to five keys at a time with one hand. We, therefore, propose to extend the input operation of PIN authentication by allowing users to type multiple numbers at a time. Figure 1 shows the difference between single-touch and multi-touch enabled PIN input schemes.

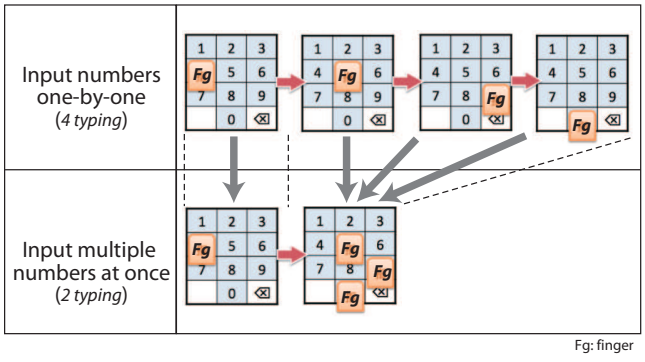

Figure 1: Comparison of single-touch and multi-touch enabled PIN input

Table 1: All input patterns in multi-touch enabled PIN authentication

input pattern(s)	# of typing(s)
****	4
(**)**, *(**)*, **(**)	3
(***)*, *(***), (**)(**)	2
(****)	1

The upper example in the figure shows a single-touch (conventional) PIN input scheme. A user presses four numeric keys one by one. The lower example, on the other hand, shows our proposed PIN input scheme. A user presses key(s) twice: A user first pushes "4" and then pushes three numeric keys "5","9" and "0" simultaneously. This means that the secret information of the proposed scheme is a combination of a PIN and an input pattern. The input pattern represents which digits of a PIN are input simultaneously. An example input pattern can be described as "∗(∗∗)∗". Each asterisk represents a number of a PIN, which is four digits in this example. Parentheses in this input pattern enclose two asterisks, and these enclosed asterisks are input simultaneously by multi-touch typing. In this example, the 1st and 4th digit are input as single numbers by single-touch typing, but the 2nd and 3rd digits are input simultaneously by multi-touch typing.

Table 1 shows all input patterns in multi-touch enabled PIN authentication. There are eight input patterns for four-digit PINs.

2.1 User Interface and Substitution Keys

Figure 2 shows the user interface of the proposed authentication for a smart phone. It is same as a standard numeric keypad except for the upper three keys: "S", "M" and "L". We call these substitution keys(SB keys).

SB keys enable users to input PINs that include the same number twice or more such as "(122)3" and "(1111)". A user can not input such PINs through a standard numeric keypad because the keypad has only one "1" or "2" key. A user, therefore, can not press two "2" keys at a time. To address this issue, we introduced the SB keys to a user interface of our scheme. We next explain how to type a PIN such as "(1111)" and "(122)3" using SB keys.

In the case of "(1111)", a user has to press four "1" keys at a time. To do this, a user presses the "1" key and the three SB keys: "S, M, and L". When one numeric key is pressed with one or more SB keys, our system interprets SB key(s) as a simultaneously pressed numeric key. This means that the three SB keys are interpreted as "1" keys at this time [*(1111)⇒(1SML)*]. The user can, therefore, type four "1" values at one time. If a PIN includes "(222)", a user has to input it by pressing the "2" numeric key and two SB keys. In this case, any combination of two SB keys is acceptable: "2" key and "S and M", "M and L" or "S and L" [*(222)⇒(2SM), (2ML) or (2SL)*].

In the case of "(122)3", a user presses two numeric keys "1" and "2" and "L" for inputting the first part of the PIN "(122)". The "L" key is interpreted as follows. SB keys "S, M and L" stand for "Small", "Medium", and "Large" respectively. When more than one numeric key is pressed along with SB key(s), our system interprets the SB key(s) using a magnitude relationship of simultaneously entered numerical values. In this example, "1" and "2" keys and the "Large" SB key were typed once. First, the system decides which number is the largest of the entered numbers because the pressed SB key was labeled "Large". In this case, "2" is the largest numeric key value. The system interprets that typing "L" is equal to typing "2". As a result, typing "(12L)" by a multi-touch typing is interpreted as "(122)".

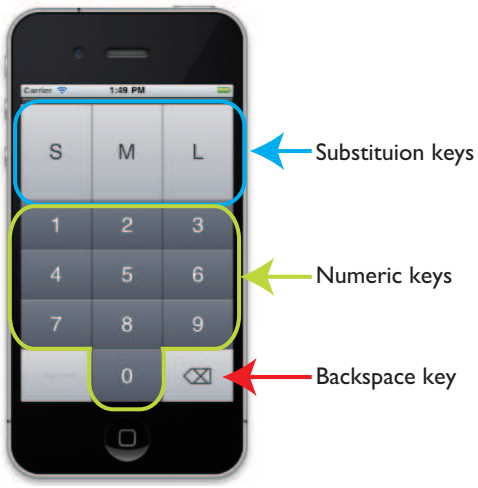

Figure 2: User interface for multiple-key typing enabled PIN authentication

We also provide some more examples of multi-touch typing with SB key(s). If a user has the PIN "(1122)", she has to press four keys simultaneously: "1" and "2" numeric keys and "S" and "L" SB keys [*(1122)⇒(12SL)*]. If a user has the PIN "(1223)", she has to input it by pressing the "1", "2" and "3" numeric keys and the "M" SB key at the same time [*(1223)⇒(123M)*]. The SB keys and this interpretation rule enables users to input any four-digit PIN with any input pattern.

3. PRELIMINARY EVALUATION

We conducted a preliminary experiment of the proposed authentication scheme. To evaluate the feasibility of the scheme, we implemented a web-based prototype system and evaluated it from the following three perspectives: a) Secret Memorability, b) Input Time, and c) Typing Errors. We used an Apple iPod Touch (4th generation model/iOS version 5.0.1) as a client terminal.

3.1 Secret memorability experiment

We conducted a secret memorability experiment over nine days with six subjects. Subjects were all males in their twenties. First, we gave the subjects an overview of our authentication scheme and its PIN input operation. We then gave them some minutes to practice using the prototype system. Each subject then freely decided a four-digit PIN and one input pattern. After that, we asked them to input their PINs to confirm them. We asked subjects to input their PINs through our system one day, four days, and nine days after setting their PINs. We allowed them to enter their PINs up to three times in each trial. This means that we judged an authentication trial failed if a subject did not succeed in entering the PIN within three input operations. This is the same condition as a bank ATM.

Table 2: Results of the secret memorability evaluation over nine days

*N*th day	1	4	9
# of failed trial	0	0	1

The results of the experiment show that all subjects succeeded at authenticating themselves each day except for one subject on the ninth day (table 2). Interviews with the subjects confirmed that none of them did anything unadvisable for keeping a secret, such as writing the PIN down. These results show that even over a short period, the secret information of the proposed scheme is not too complex to keep it in one's memory.

3.2 Input Time

We conducted a PIN input experiment of the multi-touch typing scheme with six subjects. Subjects were again all males in their twenties. We randomly generated five pairs of four-digit numbers and an input pattern. Their input patterns included at least one multiple-key input. We then asked subjects to enter them by two input schemes: one is single-touch typing and the other is multi-touch typing with a specified input pattern. Each subject, therefore, inputs a PIN ten times (five PIN numbers × two input schemes). Input time was measured by the system. We notice that the input time was not a pure length of a time during a PIN input. The measurement started when a user finished to enter a user account and ended when an authentication result was displayed in a client terminal.

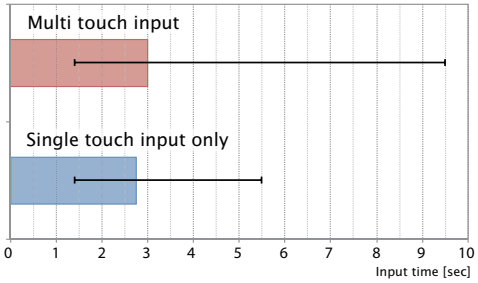

Figure 3: Results of PIN input times for single-touch and multi-touch input

Figure 3 compares results of PIN input times. The bar chart in the figure represents the average input times for both input schemes, and horizontal lines shows the range of input times. The average input times for single- and a multi-touch typing are 2.74 and 3.00 seconds, respectively. Left and right edges of the horizontal lines show the minimum and maximum times, respectively. Although the two-tailed paired t-test shows a significant difference between single- and multi-touch input, we expect that this difference in input times may shrink if users become accustomed to the proposed PIN input scheme.

4. DISCUSSION

We first discuss about the security of the proposed authentication scheme. The secret information of the proposed scheme is a combination of a four-digit number and an input pattern. The security level would, therefore, be presumed to be calculated by multiplying "the number of variations in four-digit numbers" by "the number of variations of possible input patterns". However, this presumption is incorrect because "(12)" and "(21)" are input the same way in multi-touch input. This means that the number of variations in multi-touch input is a combination, not permutation, of typed numbers. Moreover, we allow users to choose a PIN such as "(112)" or "(7766)", The number of variations in the multi-touch typing part must be calculated by using the repeated combination of typed numbers.

Table 3 shows the number of secret variations in each input pattern with a four-digit PIN. The input pattern (α) means the conventional PIN input scheme (single-touch input), and the other patterns represent one or two multi-touch typing included input patterns. From table 3, the total number of variations in the proposed scheme is 34,640. If PIN authentication allows to enter a PIN using our multi-touch input scheme, the theoretical security level of a four-digit PIN is almost three and a half times than that of the single-touch input scheme.

Table 3: Number of PIN variations in each PIN input pattern

	Input pattern	# of secret variations
(α)	****	10,000
(β)	(**)**	5,500
(γ)	*(**)*	
(δ)	**(**)	
(ϵ)	(**)(**)	3,025
(ζ)	(***)*	2,200
(η)	*(***)	
(θ)	(****)	715

Our goal is to realize a better PIN authentication with minimal modification of the conventional PIN authentication. In this paper, we extended an input operation from pressing keys one by one to pressing multiple keys at a time. The extension, however, required a small modification in both the PIN and the user interface. We do not think that these modifications significantly bothered users, and results of the user study confirmed that the proposed scheme has reasonable feasibility. We, of course, need to conduct a formal user study over a longer period.

There are some issues that should be addressed in future works. First is that the proposed scheme assumes the use of two hands. One hand holds a device, and the other hand inputs the PIN. Though a PIN can be input with one handed if a user puts a mobile device on a table, this causes an operational issue in some mobile contexts when a user can use only one hand for an input operation.

The next issue is that some combinations of a PIN and an input pattern are hard to input with multi-touch typing through a standard numeric key layout. In the case of "(578)", three fingers must push three numeric keys in a very small area of a software keyboard. In the case of "(138)", the three numeric keys are hard to push simultaneously considering the motion range of fingers of one hand. In typing these PINs, therefore, one or more of fingers can not touch a screen properly enough to push the key(s). This is a major cause of typing errors in multi-touch typing. We consider

that the use of a physical keyboard or a mobile device with a larger screen could overcome the above issues. However, the above methods are hard to apply to smaller mobile devices. We need to consider another key layout as an alternative solution to the above problems for mobile devices.

The last issue is that the proposed scheme is vulnerable to some well-known attacks such as a shoulder surfing attack and a smudge attack[1]. An attacker can identify a PIN by peeping at the input operation even if the PIN requires multi-touch typing. Even if an attacker does not see the input operation, he can look at a smudge left on the screen of a mobile device to guess or identify the legitimate user's PIN. A simple way to be more secure against these attacks is to randomize key layout in each PIN input. This is still vulnerable to an observation attack in which the input operation is captured on a video camera. Considering measures against these attacks is another future work.

We can say that our scheme integrates two secret inputs into one operation. We expect that this feature will not put much additional operational load on to users because they will perceive our scheme to be just a modified version of conventional PIN authentication. Our scheme actually modifies not only the input operation but also the user interface and the secret information in PIN authentication. These modifications add something to conventional PIN authentication: The input operation is modified by adding a multi-touch typing scheme, the user interface by adding three substitution keys, and the secret information by adding a new input pattern for a PIN. We could say that this is an extended PIN authentication and that the most obvious difference is inputting a PIN by multi-touch typing.

The above features make it possible to choose which authentication schemes to use depending on the situation. The research of Riva et al.[8] suggests that we should choose different authentication on the basis of the risk in each situation. We think that our scheme could fit this suggestion because users can change the authentication scheme in accordance with situational risk. The choice is just to change the input operation without changing the user interface or secret information drastically. This is desirable for the authentication system on a mobile device. We can offer our authentication scheme as a two-factor authentication if a system enables users to input a pattern as a one-time secret each time a user authenticates herself by our scheme. In this operation, a memory load for a user is the same as in conventional PIN authentication because she does not need to memorize an input pattern.

Here, we discuss related works. Some authentication systems use multi-touch input[2, 6, 7]. Their purposes, however, differ from that of our system. One is a countermeasure against a shoulder surfing attack for an interactive tabletop system, and the other is an alternative input scheme for the visually impaired. Various authentication systems have been proposed for improving PIN authentication[3, 4, 5]. However, their secrets and/or input operations are changed significantly from those of original PIN authentication. We guess that mobile users would feel these changes to be unnecessarily burdensome in terms of learning a new, more complicated operation and memorizing a different type of secret information, so they would hesitate to start using these alternative authentication methods.

5. CONCLUSION

In this paper, we proposed a novel user authentication scheme by extending conventional PIN authentication. This alternative PIN authentication is proposed in order to encourage mobile device users to use a more secure authentication system. The extension is that the proposed scheme allows users to input multiple numbers simultaneously instead of typing numbers one by one. We then described the extended input operation and the user interface modification, which is the addition of three substitution keys above the standard numeric keys. We also discussed about the security improvement of the proposed scheme.

We conducted a small user study to evaluate our proposal. Five out of six subjects succeeded at authenticating themselves over nine days, showing the alternative secret information to be memorizable. However, the result also showed that the input time in multi-touch input takes longer than single-touch input. We consider that the proposed scheme has the potential to provide a shorter input time when users become accustomed to it. In future works, a formal user study is needed to clarify the feasibility of the proposed scheme. We must also consider effective countermeasures against both observation and smudge attacks[1].

6. REFERENCES

[1] A. J. Aviv, K. Gibson, E. Mossop, M. Blaze, and J. M. Smith. Smudge attacks on smartphone touch screens. In *Proc. of the 4th USENIX conference on Offensive technologies*, WOOT'10, pages 1–7, 2010.

[2] S. Azenkot, K. Rector, R. Ladner, and J. Wobbrock. Passchords: secure multi-touch authentication for blind people. In *Proc. of the 14th int'l ACM SIGACCESS conference on Computers and accessibility*, ASSETS '12, pages 159–166, 2012.

[3] M. K. Chong, G. Marsden, and H. Gellersen. Gesturepin: using discrete gestures for associating mobile devices. In *Proc. of the 12th int'l conference on Human computer interaction with mobile devices and services*, MobileHCI '10, pages 261–264, 2010.

[4] A. De Luca, K. Hertzschuch, and H. Hussmann. Colorpin: securing pin entry through indirect input. In *Proc. of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 1103–1106, 2010.

[5] P. Dunphy and J. Yan. Is facepin secure and usable? In *Proc. of the 3rd symposium on Usable privacy and security*, SOUPS '07, pages 165–166, 2007.

[6] D. Kim, P. Dunphy, P. Briggs, J. Hook, J. W. Ni cholson, J. Nicholson, and P. Olivier. Multi-touch authentication on tabletops. In *Proc. of the SIGCHI Conf. on Human Factors in Computing Systems*, CHI '10, pages 1093–1102, 2010.

[7] D. Ritter, F. Schaub, M. Walch, and M. Weber. Miba: multitouch image-based authentication on smartphones. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, pages 787–792, 2013.

[8] O. Riva, C. Qin, K. Strauss, and D. i. Lymberopoulos. Progressive authentication: Deciding when to authenticate on mobile phones. In *Proc. of the 21st USENIX Security Symposium*, USENIX Security '12, pages 165–166, 2012.