fakePointer: An Authentication Scheme for a Better Security against a Peeping Attack by a Video Camera

Tetsuji TAKADA
National Institute of Advanced Industrial Science and Technology
2-41-6, Aomi, Koto-ku, Tokyo, 135-0064, JAPAN
zetaka [atmark] computer [dot] org

Abstract

Peeping attack in the real world is one of threats to a user authentication. What is worse is that an emerging attack method such as video capturing makes traditional measures against peeping attack insufficient. In this paper, I propose a unique user authentication scheme named "fakePointer" for a solution to a peeping attack by video capturing. It makes hard for attackers to get a secret even if he/she captures an authentication scene using a video camera. The fakePointer has two unique features to ensure a security against such a peeping attack. One is that fakePointer provides a double-layered interface for a secret input. This interface makes it hard for attackers to identify a legitimate user's secret even if they had a video record about target user's authentication action. The other feature is that the fakePointer uses two secrets. One is a fixed secret and the other is a disposal secret. This feature enables to change a secret input operation in each authentication. This is also a necessary feature for ensuring security because if an attacker has many video records about a same user, an attacker can extract a secret by statistical analysis.

1. Introduction

Peeping attack in the real world is one of threats to a user authentication. It is a well-known attack method that has been called as a "shoulder surfing attack". The attack is that an attacker steals a target user's secret by looking into a her/his authentication action. I have to say that the major cause of the attack is an unwise user interface because it forces users to point or type a secret directly and it enables attackers to identify a secret visually. Even more worse, in recent days, the threat has been magnified by emerging a new attack method. An attacker starts using a video camera to capture an authentication action and extracts a target user's secret from the video record later. And such incidents have already been reported[1].

I developed a novel authentication scheme named "fakePointer" as a solution to the threat. Using the fakePointer, a legitimate user does not leak her/his secret even if an attacker captures a video record about an authentication action. The fakePointer introduces two features to realize a security against the threat. First feature is a double-layered user interface for a secret input. This user interface makes hard for attackers to identify a secret visually. Second feature is that the fakePointer makes use of two secrets. One secret is a fixed secret. This is a same with a traditional authentication. The other secret is a disposal one-time secret named "answer indicator". This is also a necessary feature in order for ensuring a security against an another potential threat that would result from a video capturing. In the fakePointer, as far as a user changes an answer indicator before each authentication trial, even if a user keeps to use a fixed secret, a secret input operation is randomized and it seems to input a random number. This makes hard for attackers to extract a secret by statistical analysis even if they have multiple video records about a same user. And these features realize a more secure authentication against a peeping attack with a video camera.

2. Present State of a Peeping Attack

Peeping attack in the real world is also called a "shoulder surfing attack". An attacker stands behind of a target user closely and looks into an authentication action to steal her/his secret. I know, of course, that there are some measures against this attack such as a privacy filter and they are effective measures to some extent. A new attack method, however, emerges and it changes its threat model. An attacker starts to use a video camera to capture an authentication action in a new method. And even more worse, this attack method has been used in actual incidents around the world like ATM Scam[1]. This method greatly enhances attacker's ability to steal a secret because all the neces-

sary data for extracting a secret captures automatically and she/he can analyze the data with sufficient time, and then, extracts a secret after the data was captured.

I consider that it is a critical issue for a ubiquitous computing because of following two reasons. First reason is that the goal of a ubiquitous computing is that "the enhancing computer use by making many computers available throughout the physical environment"[2]. This means that a user becomes unavoidable that a user would have to authenticate oneself in a place where there are many eyes of other people. In a ubiquitous computing, a user will use computers that are embedded in a physical environment. It means that a user has to authenticates onself then and there if required. The other reason is that some of ubiquitous computing projects need a video camera as an environmental infrastructure. This may make a peeping attack more feasible than the current. I guess that an actual surrounding environment will be getting worse because of the following reasons.

- In ubiquitous computing projects such as a smart home, researchers have pushed forward to make an environment that a lot of video cameras are installed. The other reason for installing a video camera in an environment is a crime-prevention purpose. It might make easier for an attacker to put a video camera for malicious purpose to an environment and it may become hard to distinguish such a camera from a camera for a legitimate purpose.

- In the above environment, a bad guy may look into your authentication action through a surveillance camera and an authorized person misuses a video record for a malicious purpose. In this situation, even if a user uses small screen terminals like a mobile phone or a PDA, it has a risk to capture a video record about an authentication action through such cameras. Ubiquitous systems and appliances are often located in a busy public spaces. It would be a better condition for an attacker to do a peeping attack by both a video capturing and a human prying.

- Many people could become an attacker because they have a digital camera or a mobile phone that have a video capturing function. A downsizing of such devices also increases a risk because it becomes unclear who is actually video recording.

These considerations mean that a ubiquitous computing would make hard for us to get a place without a video camera or the eyes of other people. We, therefore, need a secure user authentication scheme that does not easily leak a secret even if an attacker captures a video record about a legitimate user's authentication action. We should, at least, have an another optional scheme as well as a traditional authentication scheme for a self-protection. And I think that it becomes more secure authentication if a novel authentication scheme combines both traditional measures and a video capture resilient scheme.

Along with the spread of the ubiquitous computing, an authentication scheme with sufficient security is needed even if an attacker gets a video record of an authentication action. I clarify a threat model that results from a peeping attack by video capturing. A first requisite against such attack is to make hard for attackers to identify a secret from a video record. It is an obvious requisite because they have a video record that was taken both a screen and an operation of an authentication. There is, moreover, one more requisite. A second requisite is that a peeping attack resilient authentication scheme is hard to extract a secret by analizing more than one video records. Since a spy camera to capture an authentication action is sophisticatedly concealed, the camera has often been put there for a certain period of time. This enables for attackers to capture video records of multiple authentication sessions in a same user because a user tends to use a same authentication machine like a bank ATM. From the consideration, a peeping attack with video camera resilient authentication scheme makes it hard to extract a secret even if attackers have more than one authentication video records about a same user and enough time to analyze them.

In this paper, I assumed following two requirements to ensure a security against a peeping attack by video capturing.

1. It must be hard for attackers to extract a secret from a video record about a legitimate user's authentication action.

2. It must be hard for attackers to extract a secret even if they have more than one video records about a same user.

3. Related Works

There are some measures against a peeping attack in both commercial products and research proposals. Before start discussion about related works, I make it clear that a peeping attack on a network, namely wiretapping, is out of the consideration in this paper.

We know that there are various ways to cover a screen and an input interface (such as a keyboard) to make it invisible from attackers. This includes a privacy filter/film, a physical cover[3] and a cloth[4]. An another popular measure is a software keyboard that a system randomly changes a key layout. Many bank ATM in Japan has this function.

There are some systems for a measure of the attack as research results. Matsumoto proposed a human-computer

cryptography method[6]. This is a challenge-response authentication scheme with consideration to a human executability. Volker et al. proposed a secure personal identification number(PIN) entry method against peeping attack[7] (figure 1). In this method, the authentication system pro-

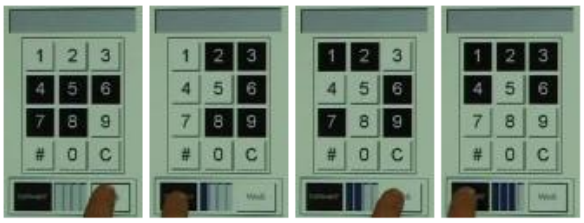

Figure 1. A Secure PIN entry method[7]

vides user with a numeric keypad that a background color of the keys is painted either black or white. These background colors are determined by the system and randomly changed in each PIN input. This method, therefore, also becomes challenge-response authentication scheme. In order to input her/his PIN, a user answers a background color of a number key of her/his PIN. Note that, in this scheme, a user answers a background color four times in order to input one digit of a PIN. Figure 1 represents an answer example of inputting a number "3". This means that a user has to type an answer 16 times in order to input 4 digits of a PIN.

Desney et al. proposed a spy-resistant keyboard for a public touch screen display[8]. An answer input operation

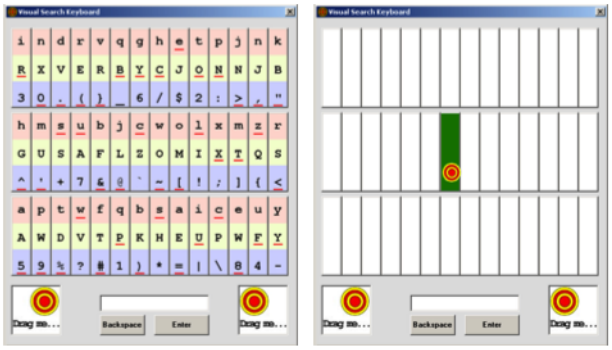

Figure 2. Spy-resistant keyboard[8]

of the system is a bit complicated. The system assigns three characters to each key. At first, a user finds a key that includes a character that a user wants to input. A user, then, has to specify an input character between three characters in the key. One of characters in each key is drawn an underline and a user can move the underline circularly by a key operation. Next, a user moves a circle mark to the key by drag and drop operation. There are two circle marks both left and right side of the bottom in a screen. A user can use either one. When a user starts dragging the mark, the system wipes out all characters from the all keys (figure 2right). Therefore, in order to identify an input character in this system, an attacker has to remember a layout of all characters and which characters are selected in each key.

The critical issue in these scheme is that there is no measure that meets previously determined requirements for a peeping attack with video camera resilient authentcation scheme. Even if legitimate users use proposed methods in the paper[7] or [8], an attacker can identify a secret from a video record. The method in the paper[6] is also vulnerable to the attack because if an attacker has many video records about a same user's authentication action, it is possible for her/him to extract a secret by statistical analysis. This means that a peeping attack by a video capturing is quite different from a shoulder surfing attack by human.

4. A Concept of the fakePointer

I explain a basic concept of the fakePointer using a safety box. A safety box has a dial for a secret number input and one marker for pointing to the number(figure 3 left). It is clearly vulnerable to a focus attack because an attacker can identify an input number visually. In order to make visual identification hard, I put multiple markers around a dial so as to be always selected all numbers of a dial (figure 3 right).

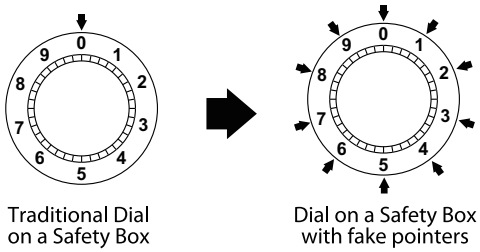

Figure 3. A Dial on a Safety Box with and without fakePointers

Before start using a safety box, an owner has to set a secret number(PIN) and also determine one fixed marker as an answer indicator. It is not secure against assumed threat in the section 2 because a dial position is always same in a secret input. It means that an attacker can not identify a fixed secret but she/he can pass an authentication by remembering a dial movement. It could say the same thing even if a user uses fixed multiple markers for a secret input.

To realize a better security, there is an another idea that a system makes a number sequence on a dial randomized in each authentication trial. It is, however, still vulnerable to the second assumed threat. If an attacker has some video records about same user's authentication action, she/he may

correctly extract a target's secret. The way is that an attacker lists all secret numbers at each marker from the video records and extracts a frequency of appearance of the numbers in each marker. Then, a number with most high appearance frequency would be a target's secret number. The reason why such attack is possible is that a user uses a fixed marker for a secret input in a consecutive authentication trials.

fakePointer, therefore, introduces randomized answer input operation to ensure a security against the second threat. A user determines a disposal secret in addition to a fixed secret before each authentication trial and uses it for a secret input. fakePointer also eliminates an assumption that a user uses only one marker for a secret input. It means that a user may use four markers to input four digits PIN and the markers are changed in each authentication.

5. How to use the fakePointer

I explain a user operation procedure of the fakePointer. I pick up a case that a user needs to withdraw a money through a bank ATM as an example. In this example, I put two assumptions. One is that a user has a 4-digit PIN as a fixed secret and the other is that a user has a mobile phone that can browse a web page.

Whenever a user needs to withdraw some money, she/he sets an "answer indicator" before an authentication at a bank ATM. The answer indicator is an essential information for a PIN input. A user can set it by using a mobile phone through a web interface. It is important point that a user should do it at a private space because a user determines an another secret by this activity.

Figure 4 represents two samples of an answer indicator. Both examples consist of 4 figures because a user inputs a number 4 times. A user can composed it from two to four types of figures. Upper example in the figure 4 is composed of four types of figures and lower example is composed of two types of figures. The fakePointer does not allow users to use an answer indicator by one figure for a security reason. The reason is clear. If it is allowed, a user tends to use such type of an indicator for a ease of memory. It, however, becomes to guess a PIN easily by attackers. A user can freely decide the number of figure types from two to four.

After this step, a user has two secrets, namely a PIN and an answer indicator, in a memory. Then a user finishes to prepare for fakePointer authentication. A user goes to a bank ATM and insert your bank account card to it. A user, then, looks at a secret input interface like figure 5. This user interface is composed of two-layered display. Numeric keys are displayed in the upper layer and figures are displayed in the lower layer. The figures in the lower layer are drawn as a background image of each numeric key and they are candidates of an answer indicator. And a layout of figures

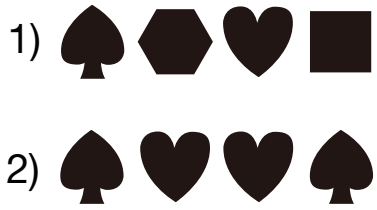

Figure 4. An answer indicator sample

Figure 5. A user interface of the fakePointer

are randomly determined by the system and are changed in each number input.

In this interface, a user can shift a numeric key layout circularly by using right and left arrow keys (figure 6). Using

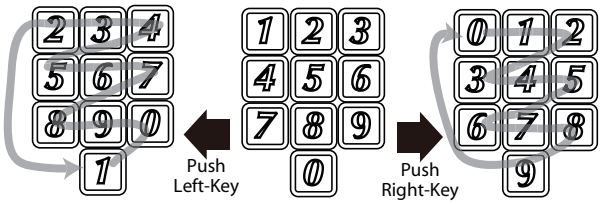

Figure 6. Numeric key layout is changed by a user operation

this function, a user has to move numeric keys until 1st digit of a PIN overlaps 1st figure of an answer indicator, and then, types a space key. As a result of it, 1st digit of a PIN is inputted to a system. In other words, a user selects **N**th digit of a PIN by **N**th figure in an answer indicator. For example, I suppose that an answer indicator in this authentication trial is the upper example in figure 4. a user changed numeric key layout and typed space key at a screen state as

figure 5 for inputting a second digit of a PIN, a user inputs a number "4" because second figure in an answer indicator is hexagonal shape in the upper sample of the figure 4 and the hexagonal shape becomes a background image of the number "4" at the figure 5. For this reason, the number of figures in an answer indicator is same with the number of digits in a PIN. A user, of course, has to repeat this operation until it finishes to input all digits of a PIN. After a secret input is complete, a user gets an authentication result.

6. Considerations

I describe about advantages and future works of the fakePointer in this section.

I consider that fakePointer is a unique authentication scheme to realize a better security against a peeping attack with a video camera. As far as a user handles an answer indicator properly, even if an attacker gets a video record of a target user's authentication action, the probability value that an attacker accidentally succeeds to impersonate someone is $1/10,000$ in 4 digit PIN authentication. It is a dramatic improvement because there was no measures against peeping attack by a video capturing. There is only one case that an attacker succeeds to get a legitimate user's secret in the fakePointer. The case is that an attacker got both an authentication video record and an answer indicator for that authentication. This means that fakePointer does not leak a user's secret even if an attacker got either one of the two.

There are two recommendations for handling an answer indicator. One is that a user has to set or update an answer indicator at a private space. Although a leakage of an answer indicator does not lead to an impersonation by an attacker, as far as an attacker does not know a fixed secret, a user should keep it secret as far as possible. I strongly recommend that a place where a user sets an answer indicator is separated a certain distance from a place where a user actually authenticates oneself.

The other recommendation is that a user updates an answer indicator in each authentication trial. The reason is that if a user continues to use a same answer indicator for a while and, on the other side, an attacker has collected some video records about his/her authentication action in that period, an attacker will succeed to extract a his/her PIN by statistical analysis. I understand that there is a trade-off between a memorability of an answer indicator and a usability of the system. From the system designer's perspective, a user has to set and use a different answer indicator in each authentication trial. By doing this, she/he can get a better security against the assumed threats.

The fakePointer has two advantages in a usability aspect. One is a simple input operation compared with other proposed systems. A user can input a secret using just only three keys (right, left arrow keys and space key). This means that fakePointer is easily operable by a terminal which is equivalent to a mobile phone and a PDA. If a terminal has a cross shaped arrow key, it is sufficient for an operation. I consider that if a terminal is equipped an interface for a scroll function like a scroll wheel or a touch panel, it could make an operation more intuitive. Figure 7 is a snapshot of a web-based prototype of the fakePointer. I will try to implement the fakePointer as an iPhone application. The

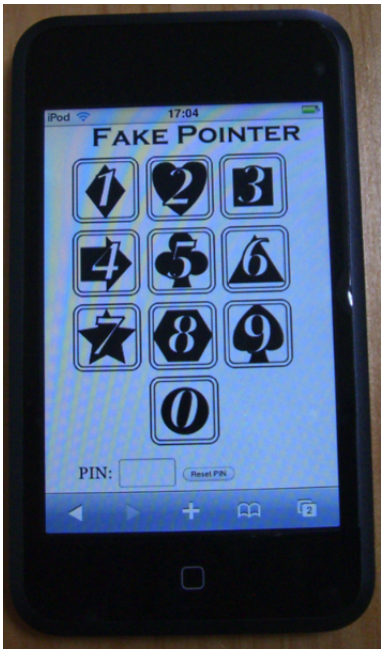

Figure 7. A Web-based prototype system of the fakePointer

other usability advantage is that an additional memory burden to users is well-controlled to a minimum increase. An additional memory burden in the fakePointer is to remember an answer indicator. However, the period when a user has to keep it in a memory is limited to a short term. The period starts from a time that a user sets an answer indicator and ends to a time that a user finishes an authentication. A user, therefore, does not have to keep an answer indicator permanently.

Moreover, A user will have no trouble even if a user forgets an answer indicator before finishing authenticate oneself. The reason is that a user can update an answer indicator at any time. It means that a user can revoke an old indicator and overwrite it to a new indicator. If an authentication system needs a security against DoS attack such that an attacker randomly updates someone's answer indicator, it puts some measure against it such as an access control by a bank account card. A user, therefore, has no reason to be afraid of forgetting an answer indicator and I expect that a user updates it frequently. I also expect that a per-

sistent memory load to users in the fakePointer is not much different from a traditional PIN authentication.

This project has some future works. One is to make an answer indicator to be more easily memorable. I consider that numbers, colors, characters and drawings may become a solution. The other is that a user interface of the fakePointer could have an another choice in some aspect. For example, a concentric key layout like a dial would be a better key layout than a numeric keypad layout because it can avoid an overlap between numeric keys and answer indicators. I also need to do a user evaluation study about a security and a usability.

7. Conclusion

In this paper, I propose a novel user authentication scheme named fakePointer. This is a unique user authentication scheme that makes peeping attack with a video camera hard. Peeping attack in the real world is one of the threats to a present user authentication and users has been exposed to a risk of this attack. In recent days, a threat of this attack becomes a bigger impact because of the change of both an attack method and an environment. We, therefore, need an another authentication scheme that ensures a security even if an attacker got a video record of an authentication action.

I consider that this is a user interface issue and also make clear that there are two assumptions as threats of a peeping attack by a video capturing. And the fakePointer introduces two features to realize a security against these assumptions. One is a double-layered user interface for a secret input. The interface makes hard for attackers to identify a input value visually, even if an attacker has a video record about a target user's authentication action. The other is that fakePointer makes a secret input operation randomized by using both a fixed secret and a disposal secret. In the fakePointer, a name of the disposal secret is "answer indicator" because it is necessary to input a fixed secret. This feature makes hard for attackers to extract a secret by statistical analysis even if they have some video records about a same user's authentication action. fakePointer also has two advantages in a usability aspect. One is a simple secret input operation and the other is a small additional memory burden to users. I will do future works described above.

References

[1] Police Department at The University of Texas Austin, ATM Scam - Bank ATMs converted to steal bank customer IDs, http://www.utexas.edu/police/alerts/atm_scam/, Site accessed at May 15, 2008.

[2] Mark Weiser, Ubiquitous Computing, http://www.ubiq.com/hypertext/weiser/UbiCompHotTopics.html, Site accessed at May 15, 2008.

[3] MATSUSHITA Electric Industrial Co.Ltd., Identification Number Input Device, Japan Patent JP,2001-147763,A, (2001).

[4] Joe Malia, Design Interactions, Private Public, http://www.interaction.rca.ac.uk/people/alumni/04-06/joe-malia/projects/project3.html, Site accessed at May 15, 2008.

[5] Lorrie Cranor, Simson Garfinkel et al., Security and Usability: Designing Secure Systems that People Can Use, O'Reilly Media, Inc., August (2005)

[6] Tsutomu MATSUMOTO: Human-computer cryptography: an attempt, *In Proc. of the 3rd ACM Conference on Computer and Communication s Security*, pp.68-75, (1996)

[7] Volker Roth, Kai Richter, Rene Freidinger: A PIN entry method resilient against shoulder surfing, *In Proc. 11th ACM Conference on Computer and Communications Security*, pp.236-245, (2004)

[8] Desney S. Tan, Pedram Keyani, Mary Czerwinski: Spy-resistant keyboard: more secure password entry on public touch screen display, *OZCHI'05: Proc. of the 19th conference of the computer-human interaction special interest group(CHISIG) of Australia on Computer-human interaction*, pp.1-10, (2005)

[9] S.Wiedenbeck, J.Waters, L.Sobrado, and J.C.Birget, Design and Evaluation of a Shoulder-Surfing Resistant Graphical Password Scheme, *Proc. of Advanced Visual Interface(AVI2006)*, pp.23-26, May (2006)

[10] N.J.Hopper and M.Blum, Secure Human Identification Protocols, *Proc. of the 7th Theory and Application of Cryptology and Information Security: Advances in Cryptology*, pp.52-66, (2001)

[11] Xiang-Yang Li and Shang-Hua Teng, Practical Human-Machine Identification over Insecure Channels, *Journal of Combinatorial Optimization*, Vol.3, No.4, Kluwer Publications, (1999)