QUANTIFYING GRAPHICAL PASSWORD STRENGTH BY NEURAL NETWORKS

Yushuo Guan, Yuanxing Zhang, Kaigui Bian

Peking University
School of Electrical Engineering and Computer Sciences
China, 100871

Lin Chen

Yale University
Department of Electrical Engineering
USA

ABSTRACT

Nowadays, graphical passwords are broadly implemented in mobile devices. The diversity of images and sketches in graphical passwords greatly increases theoretical complexity, but may not reduce the risk of shoulder-surfing attacks. Especially with the development of neural networks, shoulder-surfing attacks are simplified so that graphical passwords are under more risks than ever. In this paper, we proposed a method to quantify graphical password strength against NN enabled shoulder-surfing attacks. To our best knowledge, it is the first work in related fields.

Index Terms— graphical passwords, shoulder-surfing attacks, quantitative analysis

1. INTRODUCTION

In today of the network information times, people are increasingly using graphical passwords to login their online accounts and access personal information(e.g., visit Facebook or Twitter). Graphical passwords are knowledge-based passwords, motivated by the idea that correlates images or sketches with the shared secret [1]. The simplest applications of graphical passwords are Personal Identification Number (PIN) and text passwords, directly using numbers and characters to form passwords. More advanced passwords add complex symbols into their systems, like human faces or pictures [2], [3], [4].

The diversity of pictures improves the theoretical complexity of graphical passwords, but may not reduce the risk of shoulder-surfing attacks. Shoulder-surfing attacks, also known as peeping attacks, are defined as attacks during which adversaries can observe all actions of humans on input terminals and interactions between humans (provers) and computers (verifiers) [5]. Attackers may utilize direct observation or high-resolution cameras to record passwords. The theoretical complexity of password schemes will decrease quickly after successive shoulder surfing attacks. For example, the remaining theoretical complexity of text password will decrease to $O(1)$ after attackers recorded the whole authentication process.

For many advanced graphical passwords, it is largely insufficient for attackers to crack the password merely with the records of users, since there might be mappings between the input and password. There is no standardized mechanism to find the mappings in the past, so the efficiency of shoulder-surfing attacks fluctuated wildly between different attackers. However, the revolution in neural network (NN) simplified shoulder-surfing attacks, an advantage of NNs is that they do not need hand-crafted features and can be applied directly to raw observations, so NNs are helpful to automatically learn the mappings between the input and password.

More importantly, NN makes it possible for researchers to quantify the strength of passwords against shoulder-surfing attacks. On account of wild fluctuations on the efficiency between different attackers, traditional analysis can only make qualitative analysis on password strength against shoulder-surfing attacks. Usually they made a certain questionaire in advance and collected feedback from a group of participants, which is coarse-grained and inaccurate. NN makes the attack model standardized and provides researchers with an opportunity to quantify graphical passwords strength against shoulder-surfing attacks.

In this paper, we proposed a standardized method to quantify the password strength. We first revised the definition of the password architecture and proposed a indicator to measure password strength against shoulder surfing attacks. We then built up different NN models for graphical passwords with single or multiple bits of input. We quantified four graphical password schemes in the experiments and verified the rationality of the indicator. Other researchers may build upon this approach to analyze future systems.

To our best knowledge, our work is the first to quantify the password strength against shoulder-surfing attacks. Recent work has focused on the invention of shoulder-surfing resistant authentication. Some of them proposed theoretical principles to frustrate shoulder-surfing attacks. Shujun Li [5] mentioned three basic principles: time-variant responses, uncertainty and balance. Nicholas Hopper [6] introduced formal definitions of completeness, soundness and human excutability. Extension works proposed specialized solutions in different fields. [7] and [8] made shoulder-surfing resistant schemes for augmented reality, [9] and [10] presented specific schemes for smartphones to prevent shoulder-surfing attacks. All these

works were verified by a small number of volunteers, which is less convincing than a quantitative analysis.

2. DEFINITIONS AND SYSTEM ARCHITECTURE

In this section, we first presented basic definitions with respect to an authentication, and then introduced traditional architectures in the graphical passwords systems.

2.1. Basic Definitions

The most fundamental definitions in the authentication process are authentication challenges and answers. To login, users have to enter an answer into the system (like "123456") based on a given challenge (like "Please enter your password.").

Definition 1 ***Authentication Challenge*** $\vec{c}$ ***and Challenge Space*** $\mathbb{C}$. *$\vec{c} = [x_1, x_2, ..., x_M]$, wich contains M-bit messages for computers (verifiers) to distinguish users from malicious impersonators. Challenge space contains all possible authentication challenges that one graphical password system can provide. $|\mathbb{C}| = \sum_{i=1}^{M} |S_x^i|$, where S_x^i contains all possible options for the i-th bit message of an authentication challenge.*

Definition 2 ***Authentication Answer*** $\vec{a}$ ***and Answer Space*** $\mathbb{A}$. *For a N-bit graphical password system, $\vec{a} = [y_1, y_2, ..., y_N]$, where y_i is the i-th input of users in the authentication process. ($i \in [1, N], i \in \mathbb{N}$) Answer space contains all possible authentication answers a certain user can provide: $|\mathbb{A}| = \sum_{i=1}^{N} |S_y^i|$, where S_y^i contains all possible options for the i-th input of an answer. The answer space for different users may be different.*

Given the definitions of authentication challenge and authentication answer, we can propose the definition of authentication password.

Definition 3 ***Authentication Password*** p ***and Password Space*** $\mathbb{P}$. *Passwords can be viewed as mappings from challenge space to answer space: $p : \mathbb{C} \to \mathbb{A}$. For a specific authentication challenge $\vec{c} \in \mathbb{C}$, $p(\vec{c})$ is the correct authentication answer. Password space contains all possible authentication passwords. $|\mathbb{P}| = |\mathbb{A}|^{|\mathbb{C}|}$.*

2.2. The Architecture of graphical password systems

As shown in Figure 1, a graphical password system is constituted with a verifier (computers), several provers (human users)and attackers.

A verifier has three main functions. First, the verifier has to generate a specific authentication challenge c for a prover and collect a relevant reply $a$from the prover. Then, it has to

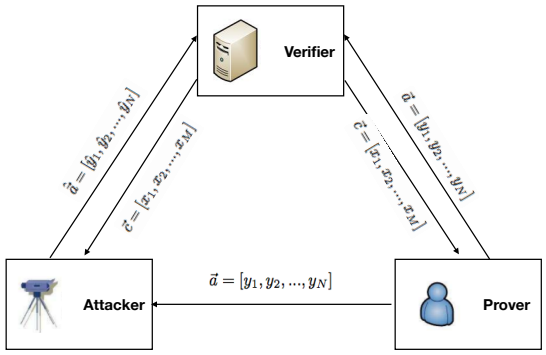

Fig. 1. The architecture of graphical password systems

memorize passwords of all provers. Besides, it is responsible for the verification of provers:

$$Verifier(a) = \begin{cases} 1, & p(c) = a \\ 0, & p(c) \neq a \end{cases}, \quad a \in \mathbb{A} \tag{1}$$

After getting an authentication challenge c from the verifier, provers calculated the answer $\vec{a} = p(\vec{c})$ and made a reply. Attackers hid in the system, recorded successive challenge and answer pairs (c, a) and trained specific NN models for different provers. The result of an attack is determined by Equation 2, and $Attacker(c, a) = 1$ will be viewed as a successful attack.

$$Attacker(c, a) = \begin{cases} 1, & NN(c) = a \\ 0, & NN(c) \neq a \end{cases} \tag{2}$$

3. DESIGN OF THE ATTACK MODEL

In this section, we described the design of the attack model. We started by explaining the training methodology. Then we described the hierarchy of the NN model and enabled it to support different graphical password systems.

The first step for the attack model is to generate a training phase. Ideally, training data would be collected with actual shoulder-surfing attacks. However, this approach is inefficient, as the training algorithm must wait until an attack finished and relevant information was recorded into the training phase. To accelerate the process, we train the attack model in a simple simulation environment that faithfully models the authentication process. The attack model's simulator maintains a verifier and a prover, which generated authentication challenges and relevant answers successively.

The hierarchy of attack models is as followed. The input is a M-dimensional vector $\vec{c} = [x_1, x_2, ..., x_M]$, showing

a simulation of an authentication challenge. The output is simply a predicted N-bit answer with respect to the authentication challenge. The input layer is simply M units, i.e. one for each bit of an authentication challenge. The hierarchy of hidden layers is optional for attackers. The output layers differed between different passwords. For the prediction of the i-bit authentication answer, if S_y^i is known by attackers in advance we will use a $|S_y^i|$-way softmax as predict the i-bit of the authentication answer. Otherwise the softmax layer will be replaced by a regression. The i-bit authentication answer can be predicted as:

$$\begin{aligned} a_i = (1 - known(S_y^i))(\sum_{i=1}^{H} w_i x_i) \\ +known(S_y^i)(\arg\max_{k \in S_y^i} \frac{exp(x_k)}{\sum_{j=1}^{S_y^i} exp(x_j)}) \end{aligned} \tag{3}$$

where H is the size of the last hidden layer, x_i is the output of the i-th unit in the last hidden layer. $known(S_y^i) = 1$ when S_y^i is known in advance, otherwise $known(S_y^i = 0$.

The NN enabled attack models only focus on the high precision rate on the test set, even if there is a high recall rate. A high precision rate on the predictions of authentication answer means attackers could pass the authentication with a higher probability. The characteristic of NN enabled attack model makes it different from traditional NN models, which pay equal attention to precision and recall rate.

With the NN enabled attack model, we could quantify passwords strength against shoulder-surfing attacks from attackers' view, since the difficulty of NN training indirectly reflects password strength against shoulder-surfing attacks. Given training set R_{train} and test set R_{test}, we defined an indicator to quantify the strength of password system $pass$ against shoulder-surfing attacks:

$$\begin{aligned} Strength_{pass|l} = \min \ \ |R_{train}| \\ s.t. \ \forall (c,a) \in R_{test}, P(Attacker(c,a) = 1 | R_{train}) \geq l \end{aligned} \tag{4}$$

The indictor $Strength_{pass|l}$ is highly influenced by the information expressivity of NN models. Therefore, for the rationality of the comparisons of password strength, the information expressivities of all NN models should be similar when $Strength_{pass|l}$ is used to quantify multiple password systems at the same time.

4. EVALUATION

In this section, we first introduce two categories of advanced graphical password systems, and present information about our dataset and model architectures. We then make two experiments and study the rationality of our indicator $Strength_{pass|l}$ and the influence of NN models' expressivity.

4.1. Experiment setup

Categories of Advanced graphical password systems: Advanced graphical passwords can be divided into two main categories based on recall and recognition. Recognition-based systems generally ask users to memorize a portfolio of images during password creation and then recognize their images from among decoys [1]. PassFaces [2] is the canonical example of recognition-based graphical passwords. PassFaces has a pool of F human face pictures, where users preselect A human faces. During login, a panel of B candidate faces is presented. Users must select the face belonging to their set from among decoys. C such rounds are repeated with different panels (generally $C < A$).

Recall-based graphical password systems are referred to as drawmetric systems [**?**]. In the recall-based systems, users recall the pattern they selected before and calculate an authentication answer based on the pattern. GrIDsure is a kind of recall-based graphical password system, which displays digits in a $D \times D$ grid. Users select and memorize a pattern consisting of an ordered subset of E grid squares in advance. On subsequent logins, digits are randomly displayed within the grid squares and users enter the sum of digits within their memorized pattern.

Setup: We made experiments on 6-bit PIN, indefinite length ($\leq$ 12) text password, PassFaces and GrIDsure [11]. 6-bit PIN has been widely used in ATM and online payment environments, and indefinite length text passwords are applied in social applications, like Facebook and Twitter. These two kinds of password systems are on behalf of traditional graphical passwords. Passfaces and GrIDsure are two canonical examples of advanced graphical passwords. In the experiments, $A = 5, B = 9, C = 1, D = 3, E = 4, F = 100$.

We use Keras to construct attack models for these four graphical passwords. Keras is a high-level neural networks API, developed with a focus on enabling fast experimentation [12]. We utilize a sequential model in our experiment. There are two hidden layers for all attack models, and each layer has an identical number of hidden units. Identical hidden layers ensure that the four NN models have similar information expressivity. The number of units in the input and output layers is determined by the size of authentication challenges and answers of different password systems, which makes little influence on the information expressivity. In the training process, we train each NN model with 10 different training sets and compute the average precision rate. We use RMSProp as the optimizer.

4.2. Experiment results

In the first experiment, we try to verify the rationality of our indicator $Strength_{pass|l}$. We record the average precision rate of four password systems when giving different sizes of the training set. All the NN models in this experiment have two hidden layers, each layer has 32 hidden units.

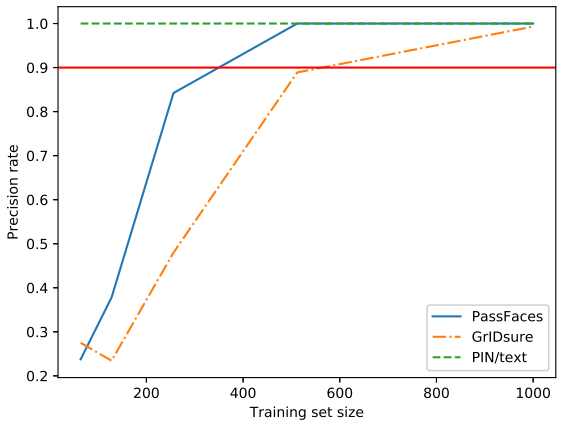

Fig. 2. The experiment results

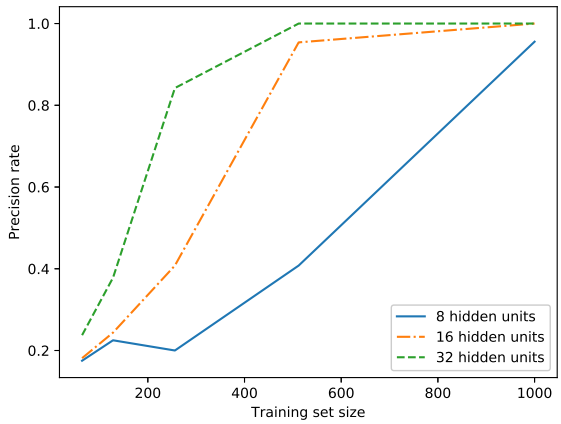

Fig. 3. The experiment results

The result is shown in Figure 2. To achieve a more than 90% accuracy, the NN model for PIN/text need really small training set, the NN model for PassFaces need 256 pieces of training data, and about 512 pieces of training data are needed for GrIDsure. The result illustrates that GrIDsure and PassFaces have greater capabilities than PIN and text passwords to protect from shoulder-surfing attacks, and GrIDsure is better than PassFaces. The experiment result is consistent with the theoretical analysis. The challenge space $\mathbb{C}$ and answer space $\mathbb{A}$ of PIN and text passwords are zero-dimension, so the authentication passwords in these systems are constant functions, which are vulnerable to shoulder-surfing attacks. For PassFaces, $|\mathbb{C}| = \binom{F}{B}, |\mathbb{A}| = \binom{A}{C}$, and the authentication password is a injective function, which is harder for NN models to crack. GrIDsure has an even larger challenge space and answer space, so it is least vulnerable to shoulder-surfing attacks.

The result also reveals vulnerability of PIN and text passwords against shoulder-surfing attacks. Nowadays, these two traditional password systems are broadly applied in mobile devices, which meet with high security risks. This is what we think the Internet industry really has to pay attention.

In the second experiment, we build up several NN models with different information expressivities and study their influence on the indicator $Strength_{pass|l}$.

Figure 3 shows the experiment results on PassFaces and GrIDsure. When there is 8 hidden units in each hidden layer, $Strength_{PassFaces|0.9}$ approaches 1000. However, $Strength_{PassFaces|0.9}$ halves when the number of hidden units increases to 32. We do not display the result on GrIDsure due to space limitations, and it shows great similarities with PassFaces on the experiment result. The attack model can not get a 90% accuracy even with 1000 pieces of training data when there are 8 hidden units each layer.

The second experiment shows that information expressivities of NN models seriously affect $Strength_{pass|l}$. The result is significant both for attackers and researchers. For attackers, the shoulder-surfing attack will be more efficient when using state-of-the-art NN models. And for researchers, the architecture of the NN model should be mentioned when quantifying a password system. Otherwise the indicator will be meaningless.

5. CONCLUSION

In this paper, we proposed a standardized method to quantify graphical password strength against NN enabled shoulder-surfing attacks. We also revised some basic definitions and presented a NN-enabled attack model. We then made experiments on four representative graphical password systems and studied the rationality of our indicator $Strength_{pass|l}$ and the influence of NN's information expressivity. In the experiments, we found traditional PIN and text passwords encountered serious security risks with shoulder-surfing attacks.

A possible direction for future work is adapting the indicator and taking time complexity into consideration.

6. REFERENCES

[1] Robert Biddle, Sonia Chiasson, and Paul C. van Oorschot, "Graphical passwords: Learning from the first twelve years," *ACM Comput. Surv.*, vol. 44, no. 4, pp. 19:1–19:41, 2012.

[2] Passfaces Corporation, "The science behind passfaces," http://www.passfaces.com/enterprise/resources/, 2009.

[3] Ian Jermyn, Alain J. Mayer, Fabian Monrose, Michael K. Reiter, and Aviel D. Rubin, "The design and analysis of graphical passwords," in *Proceedings of the 8th USENIX Security Symposium, Washington, D.C., August 23-26, 1999*, 1999.

[4] Susan Wiedenbeck, Jim Waters, Jean-Camille Birget, Alex Brodskiy, and Nasir D. Memon, "Authentication using graphical passwords: effects of tolerance and image choice," in *Proceedings of the 1st Symposium on Usable Privacy and Security, SOUPS 2005, Pittsburgh, Pennsylvania, USA, July 6-8, 2005*, 2005, pp. 1–12.

[5] Shujun Li and Heung-Yeung Shum, "Secure human-computer identification (interface) systems against peeping attacks: Sechci," *IACR Cryptology ePrint Archive*, vol. 2005, pp. 268, 2005.

[6] Nicholas J. Hopper and Manuel Blum, "Secure human identification protocols," in *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, London, UK, UK, 2001, ASIACRYPT '01, pp. 52–66, Springer-Verlag.

[7] Ruide Zhang, Ning Zhang, Changlai Du, Wenjing Lou, Y. Thomas Hou, and Yuichi Kawamoto, "Augauth: Shoulder-surfing resistant authentication for augmented reality," in *IEEE International Conference on Communications, ICC 2017, Paris, France, May 21-25, 2017*, 2017, pp. 1–6.

[8] Anindya Maiti, Murtuza Jadliwala, and Chase Weber, "Preventing shoulder surfing using randomized augmented reality keyboards," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017, Kona, Big Island, HI, USA, March 13-17, 2017*, 2017, pp. 630–635.

[9] Jia-Ning Luo and Ming-Hour Yang, "A mobile authentication system resists to shoulder-surfing attacks," *Multimedia Tools Appl.*, vol. 75, no. 22, pp. 14075–14087, 2016.

[10] Yuma Higashiyama, Naoto Yanai, Shingo Okamura, and Toru Fujiwara, "Revisiting authentication with shoulder-surfing resistance for smartphones," in *Third International Symposium on Computing and Networking, CANDAR 2015, Sapporo, Hokkaido, Japan, December 8-11, 2015*, 2015, pp. 89–95.

[11] Sacha Brostoff, Philip Inglesant, and M. Angela Sasse, "Evaluating the usability and security of a graphical one-time pin system," in *Bcs Interaction Specialist Group Conference*, 2010, pp. 88–97.

[12] François Chollet et al., "Keras," https://github.com/fchollet/keras, 2015.