First publ. in: 2010 IEEE International Conference on Communications : (ICC 2010) ; Cape Town, South Africa, 23 - 27 May 2010. Piscataway, NJ : IEEE, 2010, pp. 1- 6. ISBN 978-1-424-46402-9

Breaking Randomized Linear Generation Functions based Virtual Password System

Shujun Li*, Syed Ali Khayam†, Ahmad-Reza Sadeghi‡ and Roland Schmitz§
*Department of Computer and Information Science, University of Konstanz, Germany
†School of Electrical Engineering and Computer Science (SEECS), NUST, Islamabad, Pakistan
‡System Security Group, Ruhr-University of Bochum, Germany
§Department of Computer Science and Media, Stuttgart Media University, Germany

***Abstract*—In ICC2008 and subsequent work, Lei et al. proposed a user authentication system (virtual password system), which is claimed to be secure against identity theft attacks, including phishing, keylogging and shoulder surfing. Their authentication system is a challenge-response protocol based on a randomized linear generation function, which uses a random integer in the responses of each login session to offer security against assorted attacks.**

In this paper we show that their virtual password system is insecure and vulnerable to multiple attacks. We show that with high probability an attacker can recover an equivalent password with only two (or a few more) observed login sessions. We also give a brief survey of the related work and discuss the main challenges in designing user authentication methods secure against identity theft.

I. Introduction

In any user-oriented security system, one of the most important defensive components is the user authentication module which allows the server to grant access to legitimate users while denying access to impersonators [1]. Many types of user authentication methods exist, differing primarily in the principle used by the server to verify the identity of a claimant. Prominent authentication principles are "what you know" (e.g., secret questions, PINs, passwords), "what you possess" (e.g., token based access), and "who you are" (e.g., biometrics). There are also multi-factor user authentication methods which combine more than one principles listed above.

Despite the diversity of recently-proposed user authentication methods, the simplest one based on static passwords/PINs is still the most widely adopted method in computer and network systems. This widespread popularity of the password/PIN method is mainly a consequence of the additional costs (in terms of additional hardware and computer resources) and the usability problems that are inherent in other (more complicated) methods.

A well-known problem of a static password is its insecurity against a replay attack: static passwords can be stolen and then replayed to gain access through the server. Hence, under a static password system, stealing a user's static password implies stealing the user's identity. Typical identity theft attacks include phishing [2], malware (such as keylogger and Trojan horses) based attacks [3], and shoulder surfing attacks [4]. All of these attacks (phishing, malware and shoulder surfing) can be described by the Matsumoto-Imai threat model [5] in which the communication channel between the human user and the verifier is under the control of an adversary. These attacks are often referred to as "observer attacks", "observation attacks", or "peeping attacks" [6]. In this paper, we use the term "observer attacks".

Several solutions to combat general and specific observer attacks have been proposed since the early 1990s (see Section IV for a brief survey). The focus of this paper is the virtual password system proposed by Lei et al. [7], [8], in which a randomized linear generation function is used to protect a user from identity theft attacks. The proposed virtual password system is claimed to be secure even if an attacker can observe more than one login session. Similarly, another virtual password system was proposed in [9], which is based on the so-called "secret little functions" and codebooks. The basic idea is to use a secret function/codebook for hiding the password input from attackers. This method is very similar to some earlier solutions such as the Pass-Algorithm [10] and the codebook based solution proposed in [11].

In this paper, we re-evaluate the security of the virtual password system proposed in [7], [8] and show that its security can be easily compromised. Specifically, we show that an equivalent password can be recovered from a few (normally two) observed login sessions. This equivalent password can then be used by an attacker to impersonate the user. The computational complexity of the password cracking process is very low. We also supplement our cryptanalysis with a survey of previous work, the challenges and potential directions to provide security against observer attacks.

The rest of this paper is organized as follows. The next section briefly introduces the virtual password system proposed in [7], [8] and a recent cryptanalysis reported in [12]. In Section III we discuss how the virtual password method can be broken. Related work and more discussions are provided in Section IV. The last section concludes the paper.

II. The Virtual Password System Under Study

A. The concept of "virtual password"

The basic idea behind "virtual password" is hiding the password input using a virtual function. More specifically, the server and the user share a virtual password composed of the following two parts: 1) a fixed password $X = x_1 \cdots x_n$, where $x_i \in \mathbb{Z}$ and $\mathbb{Z}$ is the set of all password characters;

2) a virtual function $\mathcal{B} : \mathbb{Z}^{2n} \rightarrow \mathbb{Z}^n$. For each login session, the server generates a "random salt" $Y = y_1 \cdots y_n$, where $y_i \in \mathbb{Z}$, and the user has to input a "dynamic password" $K = k_1 \cdots k_n = \mathcal{B}(X, Y)$ to pass the authentication process. The fixed password is freely chosen by the user, but the virtual function is generated by the server and sent to the user for remembering. In essence, it is a common challenge-response protocol based on a shared secret: the random salt is the challenge, the dynamic password is the response, and both of them depend on the fixed password.

In [7], [8], Lei et al. proposed to use a "randomized linear generation function" as the virtual function, which can be mentally computed by human users in their brains.[1] In this case, the virtual function is actually fixed and not part of the secret any more. Lei ei al. also based their security analysis on this assumption in Section III of [7] and Section 3 of [8]. Therefore, in the following we will consider the fixed password as the only secret to be broken.

B. The virtual password system

The randomized linear generation function involved in the virtual password system is a simple linear function modulo an integer Z, where Z is the cardinality of $\mathbb{Z}$. A random integer c is generated to make the linear function behave in a random manner. The fixed password is extended to include a password string $X = x_1 \cdots x_n$ and a secret integer $a \in \mathbb{Z}$. The secret integer a is chosen such that $\gcd(a, Z) = 1$. Without loss of generality, let us assume $\mathbb{Z} = \{0, \ldots, Z-1\}$. Then, the authentication process can be described by the following steps.

- *Step 1*: The server generates a random salt $Y = y_1 \cdots y_n$ and sends it to the user, where $y_i \in \mathbb{Z}$.
- *Step 2*: The user generates a random integer $c \in \mathbb{Z}$, calculates $K = k_1 \cdots k_n$ as follows:
  - $k_1 = \mathcal{B}_1(x_1, a, y_1, c) = (ax_1 + y_1 + x_2 + c) \bmod Z$;
  - $k_i = \mathcal{B}_i(x_i, a, k_{i-1}, y_i, c) = (ak_{i-1} + y_i + x_i + c + x_{i\dot{+}1}) \bmod Z$ for $2 \leq i \leq n$, where $i \dot{+} 1 = ((i+1) \bmod n) + 1$.

  Then, the user sends K to the server.
- *Step 3*: For $c = 0, \ldots, Z-1$, the server calculates K in the same way as in Step 2, and checks if it matches the response received from the user. If no any value of c produces a match, reject the user; otherwise accept her.

Lei et al. claimed that due to the use of the random integer c, the virtual password system is secure against multiple observer attacks, i.e., even when an attacker can observe multiple login sessions, he is still not able to recover the fixed password X or the secret integer a.

C. A brute force attack reported in [12]

In [12], Coskun and Herley claimed that the above virtual password system cannot resist a brute force attack. The basic idea is as follows. For the first observed login session, the attacker can exhaustively search the four unknowns $a, c, x_i, x_{i\dot{+}1}$ for each i separately, which requires a complexity of $O(Z^4)$. Coskun and Herley expect that only $O(Z^3)$ candidates of $a, c, x_i, x_{i\dot{+}1}$ will remain at the end. This process can be done for one more observed login session, and the attacker gets $O(Z^2)$ candidates. By repeating this process for two more observed login sessions, the attacker will be able to uniquely determine values of $a, c, x_i, x_{i\dot{+}1}$. The total complexity of the brute force attack is thus $O(n(Z^4 + Z^3 + Z^2 + Z)) = O(nZ^4)$.

The above brute force attack has a shortcoming that it considers c as a fixed value for all login sessions, but in fact c is session-varying. It is possible to fix the above attack by exploiting the correlation among $k_1, \ldots, k_n$, $x_1, \ldots, x_n$, a and c, but the attack becomes more complicated and may not work for some passwords.

III. Breaking The Virtual Password System

In this section we propose a way of breaking the virtual password system under study. Our attack is not a brute force attack. It directly derives the secret integer a, from which an equivalent password string X^* can be obtained. The attacker can then use X^* and a to impersonate the user in future login attempts. Although it is not necessary to break X for impersonating the user, it may still be useful to recover the original password string X. This can be achieved via a practical brute force attack of c.

To facilitate the following description, we assume that the attacker has observed two login sessions. The observed data in the first session are $Y = y_1 \cdots y_n$, $K = k_1 \cdots k_n$, and those in the second one are $Y' = y'_1 \cdots y'_n$, $K' = k'_1 \cdots k'_n$. Denote the random integers generated by the user in the two login sessions by c and c', respectively.

In the following, we discuss how to break a, obtain X^*, and recover X. We also give a concrete example to demonstrate how the attack works, and then show some experimental results about the success rate of the proposed attack for a number of typical settings in practice.

A. Computing the secret integer a

From $k_1 = (ax_1 + y_1 + x_2 + c) \bmod Z$ and $k'_1 = (ax_1 + y'_1 + x_2 + c') \bmod Z$, we can get

$$k'_1 - k_1 = ((y'_1 - y_1) + (c' - c)) \bmod Z, \tag{1}$$

or equivalently,

$$c' - c = ((k'_1 - k_1) - (y'_1 - y_1)) \bmod Z. \tag{2}$$

For $2 \leq i \leq n$, we have the following equations:

$$k_i = (ak_{i-1} + y_i + x_i + x_{i\dot{+}1} + c) \bmod Z,$$
$$k'_i = (ak'_{i-1} + y'_i + x_i + x_{i\dot{+}1} + c') \bmod Z.$$

Subtract the first equation from the second one, we have

$$k'_i - k_i = (a(k'_{i-1} - k_{i-1}) + (y'_i - y_i) + (c' - c)) \bmod Z. \tag{3}$$

[1] In [7], [8], Lei et al. also considered using a software/hardware helper-application to do the calculation. This is not an acceptable setting in our opinion, because hardware devices are also prone to shoulder surfing and software helper-applications are prone to malware attacks.

Substituting Eq. (2) into the above equation, we get

$$\begin{aligned}&a(k'_{i-1}-k_{i-1})=\\&((k'_i-k_i)-(y'_i-y_i)-(k'_1-k_1)+(y'_1-y_1)) \bmod Z.\end{aligned}$$

In the above equation, only a is an unknown. This means that, if $\gcd(k'_{i-1}-k_{i-1}, Z)=1$, the attacker can get

$$\begin{aligned}&a=(k'_{i-1}-k_{i-1})^{-1}\cdot\\&((k'_i-k_i)-(y'_i-y_i)-(k'_1-k_1)+(y'_1-y_1)) \bmod Z,\end{aligned} \tag{4}$$

where $(k'_{i-1}-k_{i-1})^{-1}$ is the inverse of $k'_{i-1}-k_{i-1}$ modulo Z.

Observing Eqs. (1) and (3), we can see that $\gcd(k'_{i-1}-k_{i-1}, Z)=1$ is not a rare event because y'_i, y_i, c' and c are independent random variables. Assuming $k'_{i-1}-k_{i-1}$ distributes uniformly over $\mathbb{Z}$, we can get

$$\text{Prob}[\gcd(k'_{i-1}-k_{i-1}, Z)=1]=\frac{\varphi(Z)}{Z}=\prod_{p|Z}\left(1-\frac{1}{p}\right),$$

where $\varphi(\cdot)$ is Euler's totient function. When $Z=10$, the probability is $(1-1/2)(1-1/5)=2/5$, which means that $\gcd(k'_{i-1}-k_{i-1}, Z)=1$ will happen at least once for two independent login sessions with probability $1-(1-2/5)^{n-1}=1-(3/5)^{n-1}$. When $n\geq 4$, the probability is not less than $1-(3/5)^4\approx 0.87$. Therefore, in most cases two observed login sessions are enough for the attacker to uniquely determine a.

The computational complexity of this step is very low, since the only computation involved is Eq. (4), which has a worst-case complexity of $O((\log Z)^2)$ (i.e., the complexity of solving the modular inverse).

B. Obtaining an equivalent password X^*

After a is broken, the attacker can calculate an equivalent password $X^*=x_1^*\cdots x_n^*$ as follows:

- $x_1^*=ax_1+x_2+c=(k_1-y_1) \bmod Z$;
- $x_i^*=x_i+x_{i+1}+c=(k_i-ak_{i-1}-y_i) \bmod Z$ for $2\leq i\leq n$.

Note that X^* can also be calculated from K' and Y', or any other observed challenge-response pair. The computational complexity of this step is $O(n)$.

With the equivalent password $X^*=x_1^*\cdots x_n^*$ and the secret integer a, the attacker can pass any future login session by making the response in a slightly different way from Step 2 of the original challenge-response protocol.

- *Step 2**: The attacker generates a random integer $c^*\in\mathbb{Z}$, calculates $K^*=k_1^*\cdots k_n^*$ as follows:
  - $k_1^*=\mathcal{B}_1^*(x_1^*, y_1, c^*)=(x_1^*+y_1+c^*) \bmod Z$;
  - $k_i^*=\mathcal{B}_i^*(x_i^*, a, k_{i-1}^*, y_i, c^*)=(ak_{i-1}^*+y_i+x_i^*+c^*) \bmod Z$ for $2\leq i\leq n$.

  Then, the user sends K^* to the server.

It can be easily proven that the response K^* is a valid response that can help the attacker pass the user authentication in Step 3 of the challenge-response protocol:

- $k_1^*=(ax_1+y_1+x_2+(c+c^*)) \bmod Z$;
- $k_i^*=(ak_{i-1}^*+y_i+x_i+x_{i+1}+(c+c^*)) \bmod Z$ for $2\leq i\leq n$.

Apparently, the response K^* is equivalent to the response K calculated from X, a and a random integer $(c+c^*) \bmod Z$. Therefore, when the server tries $(c+c^*) \bmod Z$, it will find a match and accept the attacker as a legitimate user.

C. Recovering the original password string X

The equivalent password string X^* is different from X. In some cases, the attacker may want to recover the original password string X, in order to get some privacy-related information about the user.[2] If the user shares the same password X over multiple systems/web sites, breaking X of one system means breaking passwords of more systems.

Represent $x_1^*=ax_1+x_2+c=(k_1-y_1) \bmod Z$ and $x_i^*=x_i+x_{i+1}+c=(k_1-ak_{i-1}-y_1) \bmod Z$ in the following matrix form:

$$\begin{bmatrix}a&1&0&0&\cdots&0&0\\0&1&1&0&\cdots&0&0\\0&0&1&1&\cdots&0&0\\0&0&0&1&\cdots&0&0\\\vdots&\vdots&\vdots&\vdots&\ddots&\vdots&\vdots\\0&0&0&0&\cdots&1&1\\1&0&0&0&\cdots&0&1\end{bmatrix}\cdot\begin{bmatrix}x_1\\x_2\\x_3\\\vdots\\x_{n-1}\\x_n\end{bmatrix}=\begin{bmatrix}x_1^*-c\\x_2^*-c\\x_3^*-c\\\vdots\\x_{n-1}^*-c\\x_n^*-c\end{bmatrix} \bmod Z. \tag{5}$$

Denote the $n\times n$ square on the left side by $\mathbf{A}$, the column vector of x_i's by $\mathbf{X}$ and the column vector on the right side by $\mathbf{X}_c^*$. The above equation can be simplified to be

$$\mathbf{AX}=\mathbf{X}_c^* \bmod Z.$$

Multiplying the adjugate matrix of $\mathbf{A}$ at both sides, we have

$$|\mathbf{A}|\mathbf{X}=(\text{adj}(\mathbf{A})\mathbf{X}_c^*) \bmod Z.$$

If $\gcd(|\mathbf{A}|, Z)=1$, we immediately have

$$\mathbf{X}=(|\mathbf{A}|^{-1}\text{adj}(\mathbf{A})\mathbf{X}_c^*) \bmod Z=(\mathbf{A}^{-1}\mathbf{X}_c^*) \bmod Z.$$

If $\gcd(|\mathbf{A}|, Z)>1$, $\mathbf{X}$ cannot be uniquely solved. Instead, the modular linear system can be reduced to be

$$\frac{|\mathbf{A}|}{\gcd(|\mathbf{A}|, Z)}\mathbf{X}=(\text{adj}(\mathbf{A})\mathbf{X}_c^*) \bmod \frac{Z}{\gcd(|\mathbf{A}|, Z)}.$$

Then, we can get the value of $\mathbf{X}$ modulo $\frac{Z}{\gcd(|\mathbf{A}|, Z)}$:

$$\mathbf{X}\equiv\left(\frac{|\mathbf{A}|}{\gcd(|\mathbf{A}|, Z)}\right)^{-1}\text{adj}(\mathbf{A})\mathbf{X}_c^*\left(\bmod\frac{Z}{\gcd(|\mathbf{A}|, Z)}\right).$$

In both cases, $\mathbf{X}$ has $\gcd(|\mathbf{A}|, Z)^n\geq 1$ candidate(s) modulo Z (for each possible value of c). Since there are in total Z possible values of c, as a whole we will have $Z\cdot\gcd(|\mathbf{A}|, Z)^n$ candidates of X. If $Z\cdot\gcd(|\mathbf{A}|, Z)^n<Z^n$, Eq. (5) helps narrow down the password space.

[2] Note that many users set their passwords according to their personal information such as their birthdays.

The above process can be done independently for different observed login sessions. Then, the reduced password space can be intersected to get an even smaller password space. The reduction rate will be $r = Z \cdot \gcd(|\mathbf{A}|, Z)^n / Z^n = \gcd(|\mathbf{A}|, Z)^n / Z^{n-1}$. With m observed login sessions, the reduced size of the password space will be $Z^n r^m$. To get $Z^n r^m \leq 1$, i.e., to uniquely determine X, we have

$$m \geq \left\lceil \frac{n \log Z}{\log(1/r)} \right\rceil = \left\lceil \frac{\log Z}{(1 - 1/n) \log Z - \log(\gcd(|\mathbf{A}|, Z))} \right\rceil .$$

When $Z = 10$, $n = 4$ and $a = 3, 7, 9$, we can get $m \geq 3$. That is, only three observed login sessions are enough for the attacker to recover X.

The determinant of $\mathbf{A}$ can be derived by performing Gaussian elimination on the last row of $\mathbf{A}$ until there is only one non-zero element. The result is as follows:

$$|\mathbf{A}| = \begin{cases} a - 1, & \text{when } n \text{ is even,} \\ a + 1, & \text{when } n \text{ is odd.} \end{cases}$$

Note that X is not solvable if $Z \cdot \gcd(|\mathbf{A}|, Z)^n \geq Z^n$. For instance, when $Z = 10$, $n = 4$ and $a = 1$, $|\mathbf{A}| = 0$, which means any value of X satisfies Eq. (5), thus rendering it useless for recovering X.

The computational complexity of this step is $O\left(mnZ \cdot \gcd(|\mathbf{A}|, Z)^n + Z^2 \cdot \gcd(|\mathbf{A}|, Z)^{2n}\right)$, where the first term denotes calculation of the candidates and the second one denotes intersecting two (or more) sets of candidates.

D. An example

In this subsection, we demonstrate the proposed attack with a concrete example having the following parameters: $Z = 10$, $n = 4$, $a = 7$, $X = x_1 x_2 x_3 x_4 = 1234$.

Assume the attacker has observed two login sessions:

- $Y = y_1 y_2 y_3 y_4 = 1674$: $K = k_1 k_2 k_3 k_4 = 3526$ (corresponding to an unknown random integer $c = 3$);
- $Y' = y'_1 y'_2 y'_3 y'_4 = 6837$: $K' = k'_1 k'_2 k'_3 k'_4 = 4094$ (corresponding to an unknown random integer $c' = 9$).

First, the attacker calculates a from Eq. (4). He finds that when $i = 2$, $k'_{i-1} - k_{i-1} = k'_1 - k_1 = 4 - 3 = 1$ satisfies $\gcd(k'_1 - k_1, Z) = \gcd(1, 10) = 1$. Then, he calculates $(k'_1 - k_1)^{-1} \bmod Z = 1$. Substituting it into Eq. (4), he gets

$$\begin{aligned} a &= 1 \cdot ((k'_2 - k_2) - (y'_2 - y_2) - (k'_1 - k_1) + (y'_1 - y_1)) \bmod Z \\ &= ((0 - 5) - (8 - 6) - (4 - 3) + (6 - 1)) \bmod 10 = 7. \end{aligned}$$

Then, he calculates X^* as follows:

- $x^*_1 = (k_1 - y_1) \bmod Z = (3 - 1) \bmod 10 = 2$;
- $x^*_2 = (k_2 - ak_1 - y_2) \bmod Z = (5 - 7 \cdot 3 - 6) \bmod 10 = 8$;
- $x^*_3 = (k_3 - ak_2 - y_3) \bmod Z = (2 - 7 \cdot 5 - 7) \bmod 10 = 0$;
- $x^*_4 = (k_4 - ak_3 - y_4) \bmod Z = (6 - 7 \cdot 2 - 4) \bmod 10 = 8$.

That is, $X^* = x^*_1 x^*_2 x^*_3 x^*_4 = 2808$.

Now let us see if the attacker can impersonate the user with X^* and a. Assume in a future login attempt, the attacker gets a new random salt $Y^* = 0174$ from the server. Then, he randomly generates an integer $c^* = 6$ and calculates the dynamic password K^* as follows:

- $k^*_1 = (x^*_1 + y_1 + c^*) \bmod Z = (2 + 0 + 6) \bmod 10 = 8$;
- $k^*_2 = (ak^*_1 + y_2 + x^*_2 + c^*) \bmod Z = (7 \cdot 8 + 1 + 8 + 6) \bmod 10 = 1$;
- $k^*_3 = (ak^*_2 + y_3 + x^*_3 + c^*) \bmod Z = (7 \cdot 1 + 7 + 0 + 6) \bmod 10 = 0$;
- $k^*_4 = (ak^*_3 + y_4 + x^*_4 + c^*) \bmod Z = (7 \cdot 0 + 4 + 8 + 6) \bmod 10 = 8$.

The attacker finally sends $K^* = 8108$ to the server. The server tries different values of c and will find out that for $c = 9$ the same response 8108 can be reproduced. As a result, the attacker is accepted as the legitimate user and the attack is successful.

TABLE I
THE SUCCESS RATES OF THE PROPOSED ATTACK WHEN $n = 4$.

	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m \geq 6$
$Z = 10$	0.791	0.984	0.998	1	1
$Z = 26$	0.838	0.988	0.999	1	1
$Z = 36$	0.699	0.960	0.995	0.998	1
$Z = 52$	0.839	0.985	1	1	1
$Z = 62$	0.865	0.981	0.998	1	1
$Z = 95$	0.985	1	1	1	1

E. Experimental results

We developed MATLAB implementations of the virtual password system and of the proposed attack. We have performed a large number of simulated attacks and verified the correctness and feasibility of the attack. The code is available at http://www.hooklee.com/Papers/Data/VPS.zip. Run RandomAttack to simulate an attack, which will randomly generate a password string X and a secret integer a, and then try to crack a and derive an equivalent password string X^*. After cracking the password, a new login session is generated to show the attacker can indeed impersonate the legitimate user.

We also wrote a MATLAB function SuccessRates to estimate the real success rate of the proposed attack from a number of random attacks. For six different values of Z (corresponding to different combinations of digits, lowercase/uppercase letters, and other printable characters) and different numbers of observed login sessions, Table I gives the results we obtained from 1000 random attacks for $n = 4$. One can see that in all the cases two observed login sessions are enough to crack the virtual password system with high probability. Note that with a larger value of n, the success rate will be even higher.

IV. Related Work and More Discussions

A. Existing solutions and their limitations

In [7], [8], Lei et al. claimed their virtual password system is the first one secure against phishing, keylogging and shoulder surfing. However, there have been many other solutions proposed to these problems since the early 1990s [6], [11], [13]–[21]. Some of these solutions have been shown to be insecure, but some still remain unbroken. In the following, we give a brief survey of previous work.

Essentially, all the proposed solutions are some kind of challenge-response protocols based on shared secrets. This is not surprising because session-varying responses are essential to disable replay attacks. The main part of a design is how the user calculates the session-varying response from the challenge and the shared secret. Let the shared secret, the challenge, and the response be denoted by S, C and R, respectively. The typical design is to define a function $\mathcal{F}$ with $R = \mathcal{F}(C, S)$. Since the attacker can observe C and R, the main task is to design a function $\mathcal{F}$ such that the attacker cannot (partially or completely) recover S from C and R.

If a hardware device is available, it is not difficult to choose a cryptographically strong trapdoor one-way function, thus leading to a secure system. However, using a hardware device means that the user has to enter her secret S on the hardware device which is again prone to shoulder surfing. If the hardware device is a general-purpose one like a mobile phone or PDA, then mobile malware emerges as another potent threat [22].

In case a hardware device cannot be used, the function $\mathcal{F}$ has to be simple enough for a common user to calculate in mind. Intuitively argued, it is non-trivial to find a function $\mathcal{F}$ which is both secure and usable. Ensuring security becomes significantly more challenging if we want the system to be secure against attacks with a large number of observed login sessions. For instance, Matsumoto and Imai proposed several solutions in [5], [23], but all of them are insecure according to cryptanalysis shown in [6], [24]. Two other solutions recently proposed in [15], [21] have also been found insecure [25], [26].

Some proposed solutions [13], [14], [24], [27] remain unbroken so far, but they all suffer from the curse of usability – the average login time is too long, the user has to remember an unusually long password, or the mental computations the user has to perform are difficult. In fact, most broken solutions also have the same problem. For example, the solutions proposed in [15] ask the user to remember 30–80 images as the password, and the whole login process takes 1.5 to 3 minutes.

Since it is very difficult to design a strongly secure function $\mathcal{F}$, some solutions only aim at providing security against the weakest observer attack, shoulder surfing [16], [17], [28]–[35]. The main goal here is reduced to avoid password leaking from a few number of observed login sessions. Since the security is drastically relaxed, it is much easier to design practical solutions which are secure against shoulder surfing.

While most previous work tries to hide the password or correct responses from attackers, recently a new approach based on hidden challenges was proposed in [20], [36], [37]. The basic idea is to hide part of the challenges by using an additional channel available only to the user. For instance, in one design called UnderCover [20], the user covers her palm on a moving trackball to receive the hidden challenges from the computer. Her hand resting on the trackball obscures external observation of an attacker (a shoulder surfer or a hidden camera). The main problem with this approach is that the terminal computer must be trusted, which is not a valid assumption in some real attacks. UnderCover's dependence on the user's behavior may also introduce potential security problems since human users often do not behave in a secure manner.

B. Main challenges and potential directions

It is clear that the main challenge we are facing is the tradeoff between security and usability. It is easy to make a system secure by relaxing usability, and vice versa. The diversity of applications and human users' behavior make it very difficult to find a tradeoff acceptable for all applications and all users. It may be necessary to make the user authentication system dependent on a specific application and a specific target user group. For instance, in e-banking services, we can safely assume most users will be willing to accept a longer login time and a longer password.

Another main challenge relates to the imbalance between the human users and the potential attackers. While we assume human users have very limited computational resource – their brains, the attacker can have access to a supercomputer or even a large number of zombie computers under his control.

While a really practical solution has not been found yet, some researchers have pointed out several potential directions. For instance, in [13], Li and Shum suggested some principles and two general structures of designing challenge-response protocols secure against observer attacks – Twins and Foxtail. In a Twins protocol, the server sends a number of challenge pairs to the user, and the user is asked to make a correct response and an erroneous one for each challenge pair. In a Foxtail protocol, the user makes use of some nonlinear function to further conceal the correct response from attackers. Both protocols are general frameworks and can be implemented in many different ways, thereby offering an imminent possibility of finding an implementation with an acceptable tradeoff between security and usability in the future.

V. Conclusion

This paper re-evaluates the security of the virtual password system proposed in [7], [8], and points out that it is not secure against multiple observer attacks. With only a few number of observed login sessions, an attacker is able to break an equivalent password with high probability and uses it to impersonate the legitimate user in any future login sessions. It may also be possible to recover the original password.

How to design a practical user authentication system secure against observer attacks is still an open question, and we call for more research in this field. The main challenges include how to achieve an acceptable tradeoff between security and usability, and how to overcome the imbalance between users and attackers in terms of available computational resources.

Acknowledgments

Shujun Li was supported by a fellowship from the Zukunftskolleg ("Future College") of the University of Konstanz, Germany, which is part of the "Excellence Initiative" Program of the DFG (German Research Foundation).

References

[1] A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1996.

[2] M. Jakobsson and S. Myers, Eds., *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*. John Wiley & Sons, Inc., January 2007. [Online]. Available: http://phishing-and-countermeasures.com

[3] J. Aycock, *Computer Viruses and Malware*. Springer, 2006.

[4] Wikipedia, "Shoulder surfing (computer security)," http://en.wikipedia.org/wiki/Shoulder_surfing_(computer_security), 2009.

[5] T. Matsumoto and H. Imai, "Human identification through insecure channel," in *Advances in Cryptology – EUROCRYPT'91*, ser. Lecture Notes in Computer Science, D. Davies, Ed., vol. 547. Berlin: Springer-Verlag, 1991, pp. 409–421.

[6] S. Li and H.-Y. Shum, "Secure human-computer identification against peeping attacks (SecHCI): A survey," Technical report, 2003. [Online]. Available: http://www.hooklee.com/Papers/SecHCI.pdf

[7] M. Lei, Y. Xiao, S. V. Vrbsky, C.-C. Li, and L. Liu, "A virtual password scheme to protect passwords," in *Proceedings of IEEE International Conference on Communications (ICC'2008)*. IEEE, 2008, pp. 1536–1540.

[8] M. Lei, Y. Xiao, S. V. Vrbsky, and C.-C. Li, "Virtual password using random linear functions for on-line services, ATM machines, and pervasive computing," *Computer Communications*, vol. 31, no. 18, pp. 4367–4375, 2008.

[9] Y. Xiao, C.-C. Li, M. Lei, and S. V. Vrbsky, "Secret little functions and codebook for protecting users from password theft," in *Proceedings of IEEE International Conference on Communications (ICC'2008)*. IEEE, 2008, pp. 1525–1529.

[10] J. A. Haskett, "Pass-algorithms: A user validation scheme based on knowledge of secret algorithms," *Communications of the ACM*, vol. 27, no. 8, pp. 777–781, 1984.

[11] M. Szydlowski, C. Kruegel, and E. Kirda, "Secure input for web applications," in *Proceedings of 23rd Annual Computer Security Applications Conference (ACSAC'2007)*, 2007, pp. 375–384.

[12] B. Coskun and C. Herley, "Can 'something you know' be saved?" in *Information Security (Proceedings of ISC 2008)*, ser. Lecture Notes in Computer Science, T.-C. Wu, Ed., vol. 5222. Berlin / Heidelberg: Springer, 2008, pp. 421–440.

[13] S. Li and H.-Y. Shum, "Secure human-computer identification (interface) systems against peeping attacks: SecHCI," IACR's Cryptology ePrint Archive: Report 2005/268, http://eprint.iacr.org/2005/268, August 2005. [Online]. Available: http://www.hooklee.com/Papers/SecHCI-Survey.pdf

[14] X.-Y. Li and S.-H. Teng, "Practical human-machine identification over insecure channels," *Journal of Combinatorial Optimization*, vol. 3, no. 4, pp. 347–361, 1999.

[15] D. Weinshall, "Cognitive authentication schemes safe against spyware (short paper)," in *Proceedings of IEEE Symposium on Security and Privacy (S&P'2006)*. IEEE Computer Society, 2006, pp. 295–300.

[16] S. Wiedenbeck, J. Waters, L. Sobrado, and J.-C. Birget, "Design and evaluation of a shoulder-surfing resistant graphical password scheme," in *Proceedings of Working Conference on Advanced Visual Interfaces (AVI'2006)*. Venezia, Italy: ACM, 2006, pp. 177–184.

[17] H. Zhao and X. Li, "S3PAS: A scalable shoulder-surfing resistant textual-graphical password authentication scheme," in *Proceedings of 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'2007)*, vol. 2. IEEE Computer Society, 2007, pp. 467–472.

[18] H. Jameel, R. Shaikh, H. Lee, and S. Lee, "Human identification through image evaluation using secret predicates," in *Topics in Cryptology – CT-RSA 2007*, ser. Lecture Notes in Computer Science, vol. 4377. Berlin / Heidelberg: Springer, 2007, pp. 67–84.

[19] H. Jameel, R. Shaikh, L. Hung, Y. Wei, S. Raazi, N. Canh, S. Lee, H. Lee, Y. Son, and M. Fernandes, "Image-feature based human identification protocols on limited display devices," in *Information Security Applications (Revised Selected Papers of WISA 2008)*, ser. Lecture Notes in Computer Science, vol. 5379. Berlin / Heidelberg: Springer, 2009, pp. 211–224.

[20] H. Sasamoto, N. Christin, and E. Hayashi, "Undercover: Authentication usable in front of prying eyes," in *Proceeding of 26th Annual SIGCHI Conference on Human Factors in Computing Systems (CHI'2008)*. ACM, 2008, pp. 183–192.

[21] X. Bai, W. Gu, S. Chellappan, X. Wang, D. Xuan, and B. Ma, "PAS: predicate-based authentication services against powerful passive adversaries," in *Proceedings of Annual Computer Security Applications Conference (ACSAC'2008)*. IEEE Computer Society, 2008, pp. 433–442.

[22] K. Dunham, *Mobile Malware Attacks and Defense*. Syngress Publishing, Inc., 2008.

[23] T. Matsumoto, "Human-computer cryptography: An attempt," in *Proceedings of 3rd ACM Conference on Computer and Communications Security (CCS'96)*. ACM, 1996, pp. 68–75.

[24] C.-H. Wang, T. Hwang, and J.-J. Tsai, "On the Matsumoto and Imai's human identification scheme," in *Advances in Cryptology – EUROCRYPT'95*, ser. Lecture Notes in Computer Science, vol. 921. Berlin / Heidelberg: Springer, 1995, pp. 382–392.

[25] P. Golle and D. Wagner, "Cryptanalysis of a cognitive authentication scheme (extended abstract)," in *Proceedings of IEEE Symposium on Security and Privacy (S&P'2007)*. IEEE Computer Society, 2007, pp. 66–70.

[26] S. Li, H. J. Asghar, J. Pieprzyk, A.-R. Sadeghi, R. Schmitz, and H. Wang, "On the security of PAS (predicate-based authentication service)," in *Proceedings of 25th Annual Computer Security Applications Conference (ACSAC'2009)*. IEEE Computer Society, 2009.

[27] N. J. Hopper and M. Blum, "Secure human identification protocols," in *Advances in Cryptology – ASIACRYPT 2001*, ser. Lecture Notes in Computer Science, C. Boyd, Ed., vol. 2248. Springer-Verlag, Berlin, 2001, pp. 52–66.

[28] R. Dhamija and A. Perrig, "Déjà Vu: A user study using images for authentication," in *Proceedings of 9th USENIX Security Symposium*. USENIX Association, 2000, pp. 45–58.

[29] V. Roth, K. Richter, and R. Freidinger, "A PIN-entry method resilient against shoulder surfing," in *Proceedings of 11th ACM Conference on Computer and Communications Security (CCS'2004)*. ACM, 2004, pp. 236–245.

[30] Z. Li, Q. Sun, Y. Lian, and D. D. Giusto, "An association-based graphical password design resistant to shoulder-surfing attack," in *Proceedings of the 2005 IEEE International Conference on Multimedia and Expo (ICME'2005)*. IEEE, 2005, pp. 245–248.

[31] F. Tari, A. A. Ozok, and S. H. Holden, "A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords," in *Proceedings of 2nd Symposium on Usable Privacy and Security (SOUPS'2006)*. Pittsburgh, Pennsylvania: ACM, 2006, pp. 56–66.

[32] A. Harada, T. Isarida, T. Mizuno, and M. Nishigaki, "A user authentication system using schema of visual memory," in *Biologically Inspired Approaches to Advanced Information Technology (Proceedings of BioADIT 2006)*, ser. Lecture Notes in Computer Science, A. J. Ijspeert, T. Masuzawa, and S. Kusumoto, Eds., vol. 3853. Berlin / Heidelberg: Springer, 2006, pp. 338–345.

[33] D. Lin, P. Dunphy, P. Olivier, and J. Yan, "Graphical passwords & qualitative spatial relations," in *Proceedings of 3rd Symposium on Usable Privacy and Security (SOUPS'2007)*. Pittsburgh, Pennsylvania: ACM, 2007, pp. 161–162.

[34] E. Hayashi, R. Dhamija, N. Christin, and A. Perrig, "Use Your Illusion: Secure authentication usable anywhere," in *Proceedings of 4th Symposium on Usable Privacy and Security (SOUPS'2008)*. Pittsburgh, Pennsylvania: ACM, 2008, pp. 35–45.

[35] A. D. Luca and B. Frauendienst, "A privacy-respectful input method for public terminals," in *Proceedings of 5th Nordic Conference on Human-Computer Interaction: Building Bridges (NordiCHI'2008)*. Lund, Sweden: ACM, 2008, pp. 455–458.

[36] M. Hasegawa, N. Christin, and E. Hayashi, "New directions in multisensory authentication," in *Adjunct Proceedings of the 7th International Conference on Pervasive Computing (Pervasive'2009)*. ACM, 2009, pp. 103–106.

[37] A. D. Luca, E. von Zezschwitz, and H. Hußmann, "VibraPass - secure authentication based on shared lies," in *Proceedings of 27th International Conference on Human Factors in Computing Systems (CHI'2009)*. ACM, 2009, pp. 913–916.