Cryptanalysis of the Convex Hull Click Human Identification Protocol *

Hassan Jameel Asghar[1], Shujun Li[3], Josef Pieprzyk[1], and Huaxiong Wang[1,2]

[1] Center for Advanced Computing, Algorithms and Cryptography, Department of Computing, Faculty of Science, Macquarie University, Sydney, NSW 2109, Australia
{hasghar, josef, hwang}@science.mq.edu.au,
[2] Division of Mathematical Sciences, School of Physical & Mathematical Sciences, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore
hxwang@ntu.edu.sg,
[3] Department of Computer and Information Science, University Konstanz, Fach M697 Universitätsstraße 10, D-78457 Konstanz, Germany
hooklee@gmail.com

Abstract. Recently a convex hull based human identification protocol was proposed by Sobrado and Birget, whose steps can be performed by humans without additional aid. The main part of the protocol involves the user mentally forming a convex hull of secret icons in a set of graphical icons and then clicking randomly within this convex hull. While some rudimentary security issues of this protocol have been discussed, a comprehensive security analysis has been lacking. In this paper we analyse the security of this convex hull based protocol. In particular, we show two probabilistic attacks which reveal the user's secret after the observation of only a handful of authentication sessions. These attacks can be efficiently implemented as their time and space complexities are considerably less than brute force attack. We show that while the first attack can be mitigated through appropriately chosen values of system parameters, the second attack succeeds with a non-negligible probability even with large system parameter values which cross the threshold of usability.

Keywords: Human Identification Protocols; Observer Attack; Entity Authentication.

1 Introduction

In a human identification protocol, a human user (the prover) attempts to authenticate his/her identity to a remote computer server (the verifier). The user

* This paper has been published in *International Journal of Information Security*, vol. 12, no. 2, pp. 83-96, Springer, 2013. A short paper with the same title is to appear in *Information Security: 13th International Conference, ISC 2010, Boca Raton, FL, USA, October 25-28, 2010, Revised Selected Papers*, *Lecture Notes in Computer Science*, vol. 6531, pp. 24-30, Springer, 2011.

has an insecure computer terminal under the control of an adversary. The adversary can view the computations done at the user's terminal as well as the inputs from the user. In addition, the adversary has passive or active access to the communication channel between the user and the server. Designing a secure human identification protocol under this setting is hard, since the user can no longer rely on the computational abilities of the terminal and has to mentally perform any computations. The problem, then, is to find a secure method of identification that is not computationally intensive for humans.

Most solutions to this problem have been some form of shared-key challenge-response protocols, in which the server sends a random challenge to the user who computes a response as a function of the challenge and the shared secret. Since the server can also compute the same response, it can check whether the user's response is correct. The protocol should be able to successfully authenticate a legitimate user with high probability after a number of rounds of challenge-response messages. The adversary can always view the challenge-response pairs, since they are communicated in the open. The goal of the adversary is to impersonate the user. In order to be secure, such protocols require a function that does not "leak" too much information about the secret, in the hope that the protocol can be used for sufficiently large number of authentication sessions before the secret needs to be renewed.

In [1], Sobrado and Birget proposed a graphical human identification protocol that utilizes the properties of a convex hull. A variant of this protocol has later appeared in [2]. In [3] Wiedenbeck et al. gave a detailed description of the protocol from [1], with a usability analysis employing human participants. Since the work reported in [3] is more comprehensive, we will adhere to the protocol described therein for our security analysis in this paper. Following the term used in [3], we call the protocol Convex Hull Click or CHC in short. The protocol can be roughly described as follows: in the setup phase, the user and the server share a subset of graphical icons as a secret. As in all human identification protocols, the setup phase is assumed to take place in a secure setting, outside the reach of any adversaries. In an identification session, the server shows a screen of randomly placed graphical icons. The user mentally forms a convex hull of the secret graphical icons and then clicks a random point inside this convex hull. In this paper, we attempt to rigorously analyse the security of the CHC protocol. In particular, we describe two probabilistic attacks on the protocol and show its weaknesses against a passive eavesdropping adversary.

2 Related Work

The identification protocol of Matsumoto and Imai [4] was the first attempt at designing a human identification protocol secure under the aforementioned setting. The protocol, however, was shown to be insecure by Wang et al. [5] who proposed some fixes but which render the resulting protocol too complex to execute for most humans. Matsumoto also proposed some other protocols in [6]. However, the security of these protocols can be compromised after a few

authentication sessions [7, 8]. Some other proposals for human identification protocols that have been shown to be insecure were proposed by the authors in [9, 10, 11]. These protocols were cryptanalysed in [12, 13, 14].

Hopper and Blum proposed the well-known HB protocol, which is based on the problem of learning parity in the presence of noise [8]. To the best of our knowledge, this is the only human identification protocol whose security is loosely based on an NP-hard problem. Yet, the protocol has some weaknesses as it requires the user to send a wrong answer with a probability between 0 and 0.5, which is arguably hard for most humans. Li and Teng's protocols [15] seem impractical as they require a large size of secret (3 secrets of 20 to 40 bits). Li and Shum's protocols [7] have been designed with some principles in mind, such as using hidden responses to challenges. This loosely means that the responses sent to the server are non-linearly dependent on the actual (hidden) responses. However, the security of these protocols has not yet been thoroughly analysed. Jameel et al. [16, 17] have attempted to use the gap between human and artificial intelligence to propose two image-based protocols. The security, however, is based on unproven assumptions. Furthermore, it appears difficult to automatically generate random challenges without human intervention from the server side. More recently, Asghar, Pieprzyk and Wang have proposed a human identification protocol in [18]. The usability of the protocol is similar to Hopper and Blum's protocols. But an authentication time of about 2 to 3 minutes is still not practical.

Some proposals have been designed to be secure against a very restricted adversary; the human shoulder-surfer. The schemes from [19] and [20] are examples. The convex hull click based identification protocols from Wiedenbeck [3] and Zhao and Li [2] can also provide good security against these adversaries. A different direction is to construct alternative input devices that use different human senses to hide the challenges to or the responses from the user. This can potentially be more secure than a keyboard based input device, since the adversary's view is restricted. For example, Sasamoto et al.'s scheme UnderCover [21] uses a haptic device on which the user places his/her palm. The palm hides any external observation, while the user can receive part of the challenge from the haptic device which touches the user's palm.

3 The CHC Human Identification Protocol

We begin with the definitions of polygons and convex hulls [22].

Definition 1 (Polygon). *A* polygon *is a piece-wise linear, closed curve in a plane. The straight line segments forming the closed curve are called the* sides *of the polygon. A point joining two consecutive sides is called a* vertex*. A polygon is* simple *if it does not cross itself.*

Definition 2 (Interior, Exterior and Boundary). *The set of points in the plane that lie outside a simple polygon is called its* exterior*; the set of points lying on the polygon form its* boundary *and the set of points inside the boundary*

of the polygon is called its interior. *If a point P lies on the boundary or in the interior of a polygon, we say that the polygon* contains *P or P is contained in the polygon.*

Definition 3 (Convex Polygon). *A simple polygon is* convex *if all points on the line segment joining any two points in its boundary or interior are contained in the polygon.*

Definition 4 (Convex Hull). *The* convex hull *of a set of points Π, is the smallest convex polygon for which every point in Π is contained in the polygon.*

Figure 1 shows the convex hull of the set of points $\Pi = \{P_1, P_2, \ldots, P_7\}$. We shall denote the convex hull of a set of points Π by $\mathsf{ch}(\Pi)$. We denote the membership relation "contains" by $\in$. For instance, in Figure 1, $P_i \in \mathsf{ch}(\Pi)$, for $1 \leq i \leq 7$. The convex hull of 3 points is a triangle. Hence, we will use the terms convex hull and triangle interchangeably for the case of 3 points. The next section describes the CHC human identification protocol.

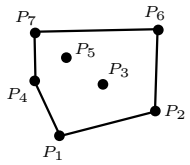

Fig. 1. The convex hull of a set of points Π.

3.1 The Protocol

Informally, we define an identification protocol as an interaction between the prover $\mathcal{H}$ and the verifier $\mathcal{C}$ such that $\mathcal{C}$ accepts $\mathcal{H}$ with high probability, if they interact using the same secret. A human identification protocol is an identification protocol in which the prover is a human who has to mentally perform the computations. In the CHC human identification protocol, initially, $\mathcal{H}$ and $\mathcal{C}$ choose k graphical icons from a set of n. These k icons constitute the shared secret between the two parties. As an example, k can be 5 and n can be 100. This is called the setup phase. When $\mathcal{H}$ wants to prove its identity to $\mathcal{C}$, the following protocol is carried out.

CHC Protocol.

1: $\mathcal{C}$ randomly samples a set of m graphical icons out of n. Here, m is a random positive integer between n and some lower bound $m_{\min}$. $\mathcal{C}$ ensures that at least 3 of the k secret icons are included in these m graphical icons. These icons are distributed randomly on the screen of the user's computer terminal within a rectangular frame and aligned in a grid.

2: $\mathcal{H}$ mentally computes the convex hull of any 3 secret icons displayed on the screen and randomly clicks a point contained in this convex hull. $\mathcal{H}$ does not need to click on the icons themselves. $\mathcal{H}$ can click anywhere on the screen in the interior or boundary of this convex hull. Notice that this is equivalent to clicking on the convex hull of all the secret icons present in the screen[4].
3: $\mathcal{C}$ repeats the process a certain number of times and accepts or rejects $\mathcal{H}$ accordingly.

□

For the ease of analysis, we make some assumptions as follows.

- Instead of choosing m randomly each time, we assume it to be fixed. In fact, we will later see that once n and k are fixed, we do not have much freedom in choosing m, if a certain attack is to be avoided.
- We replace graphical icons by non-negative integer lattice points on a real plane, enclosed within a rectangular area. The lattice points are identified by a unique integer label from the set $\{1, 2, \ldots, n\}$. Notice that, graphical icons are displayed for the ease of humans. Thus, from an analytical point of view, the two representations are equivalent. Throughout this text, we will use the terms, icons and labels, interchangeably.
- One round of the protocol will thus constitute the positive quadrant of the real plane. The m graphical icons are replaced by randomly placed integer lattice points on this quadrant, each one having a unique label. The user's set of secret icons is thus also a set of integer labels; see Figure 2. We shall call the area enclosed in the rectangle as the rectangular lattice area or simply the rectangle.

Example 1. Suppose $n = 30$ and $m = 25$. Further, suppose $k = 4$, and $\mathcal{H}$ and $\mathcal{C}$ share the secret $\{7, 15, 27, 30\}$. Figure 2 shows one run of the protocol. Since the challenge only contains $7, 15$ and 30 from the set of secret labels, $\mathcal{H}$ forms the convex hull of the points corresponding to these labels and outputs a random point contained in this convex hull. This point is depicted by the symbol $\times$ in the figure. Note that this point is not necessarily a point on the lattice. We can consider this to be a point in the real plane. That is, it belongs to $\mathbb{R}^2$. □

3.2 Description of the Adversary

The adversary considered here is a passive shoulder-surfing adversary, $\mathcal{A}$. The goal of the adversary is to impersonate $\mathcal{H}$ by initiating a new identification session with $\mathcal{C}$, after observing a few identification sessions between $\mathcal{H}$ and $\mathcal{C}$. From

[4] There can be a slight difference in the real setting. It is arguably hard for a user to click on the boundary of convex hulls with accuracy. It is easy to see that a convex hull of more than 3 secret icons contains the convex hull of any 3 secret icons. Therefore, the boundary points of the latter convex hull might be in the interior of the former. Thus, these boundary points will be easier for the user to click in the former case. To simplify our analysis in this paper, we ignore the difference between the two settings.

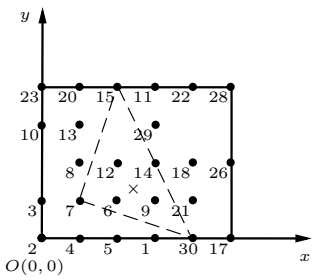

Fig. 2. One round of the convex hull protocol.

a theoretical point of view, the adversary is given a "transcript" of communication between $\mathcal{H}$ and $\mathcal{C}$ and is then allowed to play a game with $\mathcal{C}$ trying to fool $\mathcal{C}$ into accepting $\mathcal{A}$ as $\mathcal{H}$. The success of $\mathcal{A}$ is measured by the probability of successful impersonation. Notice that $\mathcal{A}$ may not need to find the shared secret between $\mathcal{H}$ and $\mathcal{C}$. For instance, if $\mathcal{A}$ randomly clicks anywhere in the challenge, then there is a possibility, albeit very small, that it can successfully impersonate $\mathcal{H}$. It is assumed that the adversary cannot view the setup phase of the protocol, i.e., when $\mathcal{C}$ and $\mathcal{H}$ establish a shared secret. However, every subsequent identification session can be viewed by the adversary.

Each identification session between $\mathcal{H}$ and $\mathcal{C}$ consists of a fixed number of challenge-response pairs. A challenge is a screen full of graphical icons (or labels) and the corresponding response is a point $P \in \mathbb{R}^2$. The number of challenge-response pairs in an authentication session is chosen such that the probability of $\mathcal{A}$ impersonating $\mathcal{H}$ with random clicks is very small. We denote this fixed number by r_0. For example, Wiedenbeck et al. used $r_0 = 10$ [3, §6, pp. 183]. They also mentioned that the implementation of their protocol ensures that convex hulls of secret icons occupying more than half the screen are rare[5] [3, §3, pp. 180]. Thus, we can assume that the average probability of success of $\mathcal{A}$ in impersonating $\mathcal{H}$, through random clicks, is less than $(\frac{1}{2})^{r_0}$.

4 Attack 1: Difference in Distributions

Our first observation is that $\mathcal{C}$ has to ensure that at least 3 out of k secret labels are displayed on the screen. There is no such restriction on the non-secret labels. Naturally, this may lead to two different probabilities for the secret and

[5] If the challenge is generated as mentioned in the protocol description, then there is a chance that the probability that a random click is contained in the convex hull of secret icons is greater than 1/2. However, through our experimental results, we found that for the values of system parameters used in this paper, the probability is less than 1/2. Thus, we do not require any modifications to ensure that this probability is less than 1/2.

non-secret labels. The probabilities depend on how the random challenges are generated. There are several possible ways to generate random challenges. To simplify our discussion, in this paper we consider the following approach: first generate a random number $l \in \{3, \ldots, k\}$, then randomly select l secret labels and $m - l$ non-secret labels to form the challenge. Note that this approach was also the one adopted in the implementation from [3].

We now calculate the probability of generating a secret label and compare it with the probability of generating a non-secret label. Let N denote the set of all labels and let K denote the set of secret labels. Thus, $|N| = n$ and $|K| = k$. Let $\overline{K}$ denote the set of non-secret labels. Thus, $K \cup \overline{K} = N$. We assume that the adversary $\mathcal{A}$ has observed $r \geq 1$ challenges sent from $\mathcal{C}$ to $\mathcal{H}$. For each of these challenges, we denote the set of secret labels appearing in the challenge by K_j and the set of non-secret labels by $\overline{K}_j$, for $1 \leq j \leq r$. Notice that $|K_j \cup \overline{K}_j| = m$ for all j. In this attack, we do not even require the responses to these challenges. For $1 \leq i \leq n$ and $1 \leq j \leq r$, define the following indicator random variables:

$$S_{i,j} = \begin{cases} 1 \text{ if label } i \text{ appears in challenge } j \\ 0 \qquad\qquad\quad \text{otherwise} \end{cases}$$

Then, for any $i \in K$ and any $j \in \{1, \ldots, r\}$, we have that

$$\begin{aligned} \Pr\left[S_{i,j} = 1, i \in K\right] &= \sum\nolimits_{l=3}^{k} \Pr\left[S_{i,j} = 1 \middle| |K_j| = l\right] \Pr\left[|K_j| = l\right] \\ &= \frac{3}{k}\frac{1}{k-2} + \frac{4}{k}\frac{1}{k-2} + \cdots + \frac{k}{k}\frac{1}{k-2} \\ &= \frac{1}{k\,(k-2)}\left(\frac{k\,(k+1)}{2} - 3\right) \end{aligned} \tag{1}$$

And for any $i \in \overline{K}$, we have that

$$\begin{aligned} \Pr\left[S_{i,j} = 1, i \in \overline{K}\right] &= \sum\nolimits_{l=3}^{k} \Pr\left[S_{i,j} = 1 \middle| |\overline{K}_j| = l\right] \Pr\left[|\overline{K}_j| = l\right] \\ &= \frac{m-3}{n-k}\frac{1}{k-2} + \frac{m-4}{n-k}\frac{1}{k-2} + \cdots + \frac{m-k}{n-k}\frac{1}{k-2} \\ &= \frac{1}{k-2}\frac{1}{n-k}\left(m - 3 + m - 4 + \cdots + m - k\right) \\ &= \frac{1}{k-2}\frac{1}{n-k}\left(m\,(k-2) - \frac{k\,(k+1)}{2} + 3\right) \end{aligned} \tag{2}$$

Now, let $S_i^{(r)}$ denote the number of times label i appears in r challenges. Then,

$$E[S_i^{(r)}] = \sum\nolimits_{j=1}^{r} E\left[S_{i,j}\right]$$

Thus, for $i \in K$, we have

$$E[S_i^{(r)}, i \in K] = r \Pr\left[S_{i,j} = 1, i \in K\right]$$

And for $i \in \overline{K}$, we get

$$E[S_i^{(r)}, i \in \overline{K}] = r \Pr\left[S_{i,j} = 1, i \in \overline{K}\right]$$

Thus, the two expected values will be different, provided the two probabilities in Equations 1 and 2 are different. For instance, when $n = 112, m = 70, k = 5$ and $r = 100$, we get

$$E[S_i^{(r)}, i \in K] = (100)(0.8) = 80$$

and

$$E[S_i^{(r)}, i \in \overline{K}] = (100)(0.6168) = 61.68$$

Hence, in 100 randomly generated challenges, we expect the secret labels to appear around 80 times each and the non-secret labels to appear around 62 times each. This observation immediately leads to the following probabilistic attack.

Attack 1.

Input: r challenges.
Output: k labels.

1: Count the number of times each label appears in the r challenges.
2: Output the top k most frequently occuring labels.

□

The above algorithm has a high success rate provided the two aforementioned probabilities differ considerably. We ran simulations for two different sets of system parameter values and the results are shown in the first two rows of Table 1. For each set of values, a total number of 1000 simulated attacks were performed. As can be seen, the algorithm, on average, outputs almost all the secret labels even with only 100 given challenges. And, in both sets of values, the probability of obtaining all k secret labels as the output of Attack 1 is higher than 0.5. Since each identification session contains $r_0 = 10$ challenges, this implies only 10 identification sessions. The set of labels thus obtained can be verified against a few responses corresponding to these challenges. Once the secret labels are obtained, it is trivial for $\mathcal{A}$ to impersonate $\mathcal{H}$. To avoid this attack, the two probabilities should be equal. This gives the following lemma:

Lemma 1. *If* $\Pr[S_{i,j} = 1, i \in K] = \Pr\left[S_{i,j} = 1, i \in \overline{K}\right]$ *for* $j \in [1, r]$, *then* $n = \frac{2km}{k+3}$.

Proof. From Equations 1 and 2, we get:

$$\begin{aligned}
&\frac{1}{k-2}\frac{1}{n-k}\left(m(k-2) - \frac{k(k+1)}{2} + 3\right) = \frac{1}{k(k-2)}\left(\frac{k(k+1)}{2} - 3\right) \\
\Rightarrow\ &\frac{1}{n-k}\left(m(k-2) - \frac{k(k+1)}{2} + 3\right) = \frac{1}{k}\left(\frac{k(k+1)}{2} - 3\right) \\
\Rightarrow\ &km(k-2) - \frac{k^2(k+1)}{2} + 3k = \frac{nk(k+1)}{2} - 3n - \frac{k^2(k+1)}{2} + 3k
\end{aligned}$$

This implies that,

$$\begin{aligned} km\,(k-2) &= \frac{nk\,(k+1)}{2} - 3n \\ \Rightarrow 2km\,(k-2) &= nk^2 + nk - 6n \\ \Rightarrow 2km\,(k-2) &= n\left(k^2 + k - 6\right) \\ \Rightarrow 2km\,(k-2) &= n\,(k+3)\,(k-2) \\ \Rightarrow \frac{2km}{k+3} &= n \end{aligned}$$

□

The last two rows of Table 1 show the results of the simulations with the value of m calculated according to Lemma 1. The results are what we expect if m out of n objects are sampled at random. Thus, this fix prevents this type of attack. Notice that, if the value of m is chosen according to the equation in Lemma 1, the probability of any label appearing in a challenge is m/n. That is, all labels are equally likely to appear in a challenge. This can be verified by direct substitution. Of course the above formula does not always give an integral solution. In that case, the nearest integer value of n or m can be chosen. The resulting probability difference would be statistically small, requiring a huge number of challenges to differentiate. Alternatively, we can only look for integral solutions to the equation, for instance $n = 120, m = 90$ and $k = 6$. In this case, Attack 1 will not work no matter how many challenges are observed.

Table 1. Simulation Results for Attack 1

n	m	k	r	Average Number of Secret Labels	Probability of Finding all k Secret Labels
112	70	5	100	4.6	0.622
500	200	12	100	11.4	0.554
112	90	5	100	0.1	0.000
500	313	12	100	0.2	0.000

Readjusted values of Parameters. Wiedenbeck et al. used the values $n = 112$ and $k = 5$ for their user study. The value of m was dynamic, ranging from 43 to 112 giving an average value of 83 [3, §4.1, pp. 181]. In light of Lemma 1, for $n = 112$ and $k = 5$, we suggest $m = 90$ instead. It should be noted that this only guarantees that the system will be secure against Attack 1, since for such small values brute force attack is feasible. For high security, Wiedenbeck et al. suggest $n = 500$, $m = 200$ and $k = 12$. However, for $m = 200$ and $k = 12$, the value $n = 320$ should be used; and for $n = 500$ and $k = 12$, the value $m = 312.5 \approx 313$ should be used. This last value of m can become prohibitive, since it will most probably be hard, for an average human user, to find secret icons among a pool

of icons as large as 300. Thus, Lemma 1 limits the values of system parameters that can be used.

5 Number of Candidates Satisfying a Challenge-Response Pair

Before we proceed to the description of our second attack, we would like to try to answer the following question of theoretical interest: Given one challenge-response pair, how many convex hulls of three labels contain the response point P? For simplicity, we assume the response point P to be in $\mathbb{R}^2$. As before, if P is contained in the convex hull of the points in S, we denote it by $P \in \mathsf{ch}(S)$.

We see that each convex hull of three lattice points is a 3-combination of labels. Also, only m out of a total of n labels occur in one challenge. Therefore the number of convex hulls of three labels that contain the point P is less than or equal to $\binom{m}{3}$. We assume that all $\binom{m}{3}$ possible 3-combinations of labels are enumerated and let $\Gamma_1, \ldots, \Gamma_{\binom{m}{3}}$ denote these 3-combinations. Thus each 3-combination is a set of 3 labels. We define the indicator random variable corresponding to Γ_i by C_i, which is 1 if $P \in \mathsf{ch}(\Gamma_i)$. Let $C = \{\Gamma_i | P \in \mathsf{ch}(\Gamma_i), 1 \leq i \leq \binom{m}{3}\}$. Then, we have that:

$$E\left[|C|\middle|P\right] = \sum\nolimits_{i=1}^{\binom{m}{3}} E\left[C_i|P\right]$$

And,

$$E\left[|C|\right] = \int\limits_R E\left[|C|\middle|P\right] f_P\left(P\right) dP = \int\limits_R \left(\sum\nolimits_{i=1}^{\binom{m}{3}} E\left[C_i|P\right]\right) f_P\left(P\right) dP$$

where R denotes the rectangle and $f_P(P)$ is the probability density function of the point P. We assume that the bottom-left corner of the rectangle coincides with the origin of the xy-coordinate system. Let $(a, 0)$ and $(0, b)$ be the coordinates of the bottom-right and top-left corners of the rectangle, respectively. The area of the rectangle is therefore ab. If we assume P to be uniformly distributed over the rectangle, we get:

$$E\left[|C|\right] = \frac{1}{ab} \int\limits_R \left(\sum\nolimits_{i=1}^{\binom{m}{3}} E\left[C_i|P\right]\right) dP = \frac{1}{ab} \sum\nolimits_{i=1}^{\binom{m}{3}} A_i$$

where A_i is the area of $\mathsf{ch}(\Gamma_i)$. Now, let γ be the fraction of the number of convex hulls containing the point P. Then, it can be obtained as:

$$\gamma = \frac{E\left[|C|\right]}{\binom{m}{3}}$$

There are two things wrong with this approach. First, the bounding rectangle is chosen such that the number of lattice points it can accomodate is considerably

higher than m. Thus $(a+1)(b+1) > m$. This means that the placement of m labels will be different in different challenges, thus giving a different value of γ each time. This feature is included in the CHC protocol to enable humans to conveniently locate the secret icons. Secondly, and more importantly, the distribution of P is not uniform over the rectangle. This is true even if we assume the user to select a point uniformly at random, contained in the convex hull of the secret labels; the points around the boundary of the rectangle have a much lower probability of being chosen, as they are contained in the least number of convex hulls. Figure 3 shows the distribution of the point P in a simulation of $10,000$ challenges. We chose the parameters: $n = m = 16$, $a = 3$, $b = 3$ and $k = 3$. The point P is generated as a uniform random point contained in the convex hull of any three secret labels. We used Turk's method to compute a random point within a triangle [23] (see Appendix A). As the figure shows, the density is lower

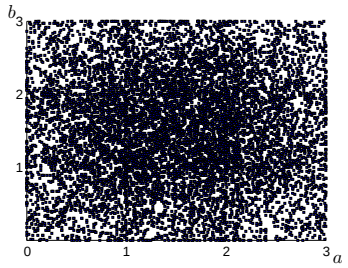

Fig. 3. The distribution of P. Notice less concentration along the boundaries of the rectangle

along the boundaries as compared to the center. For these reasons, we use an experimental approach to find the value of γ. We run the following algorithm to find an approximate value of γ. We randomly select k out of n labels as a secret to calculate the value of γ.

Algorithm Find γ.

Input: Parameters n, m, a, b and k; k secret labels and a precision value t (say $= 100$).

Output: Approximate value of γ.

1: **for** $i = 1$ to t **do**
2: Generate a random challenge as in the convex hull protocol.
3: Form a convex hull of any 3 secret labels.
4: Sample a random point P contained in this convex hull.
5: Initialize $C \leftarrow 0$.
6: For all $1 \leq j \leq \binom{m}{3}$, check whether the point P is contained in $\mathsf{ch}(\Gamma_j)$. If yes, increment C by 1.

7: $\gamma_i \leftarrow C/\binom{m}{3}$.
8: Output $\gamma = \frac{1}{t}\sum_{i=1}^{t} \gamma_i$.

□

Table 2 shows the values obtained for γ for two different sets of parameters. This gives us the result that with these parameter values, we expect approximately 15 percent of the convex hulls of 3 labels in a challenge-response pair to contain the response point. The 3-combinations making up these convex hulls are possible candidates for the secret 3-combination. While the average is around 0.15 for these choices of parameters, there was substantial deviation found in individual values with some values being as low as ≈ 0.016 and some being as high as ≈ 0.25. This is largely because different challenges have different sizes of convex hulls. As a result, in order to get accurate results, we compute the value of γ for each challenge-response pair separately. Thus $\gamma_1, \gamma_2, \ldots$, will now denote the values of γ for challenge-response pairs $1, 2, \ldots$, respectively.

Table 2. Values of γ.

Runs	t	n	m	k	a	b	Average γ
10	100	112	90	5	13	13	0.1460
5	100	160	100	12	13	13	0.1479

6 Attack 2

One attack mentioned by the authors in [3] is to find all k-combinations of the set of n labels whose convex hull does not satisfy a challenge-response pair (does not contain the response point). Initially, the list contains all k-combinations of n labels. After the observation of each challenge-response pair, the k-combinations of labels, whose convex hull does not contain the response point, are discarded from the list. The sole remaining k-combination is then the k secret labels of $\mathcal{H}$. The attack's time and space complexity is $O(\binom{n}{k})$. Thus, with the values of $n = 320$, $m = 200$ and $k = 12$, the time and space complexity of this attack is roughly 2^{70}, which can be intractable especially in terms of memory resources.

We notice that the user only has to form a convex hull of 3 labels. Thus, in theory, there could possibly be an attack of complexity $O(\binom{m}{3})$. Our second attack runs within this bound and outputs one of the k secret labels with high probability. The basic idea of the attack is as follows. We first find all candidate 3-combinations, i.e., all 3-combinations of labels whose convex hull contains the point P corresponding to a challenge-response pair. Next, we construct a frequency list that maintains the record of the number of candidate 3-combinations in which each label appears. We update the frequency list by including more challenge-response pairs and seeing if the candidate 3-combinations also satisfy these challenge-response pairs. Finally, the label that appears with

the highest frequency is the output of the attack. Section 6.3 explains in detail why the output is one of the secret labels with high probability. Once one or more secret labels are obtained, the adversary can impersonate $\mathcal{H}$. Notice that this can be done even with less than k secret labels. Section 6.4 describes how this is achieved. Note that human users tend to remember multiple icons with some hints. It is quite likely that they will select all secret icons that belong to the same category, e.g., icons of software or national flags. In this case, revealing part of the secret will lead to a better guess of the whole set of secret icons.

6.1 The Attack

We now describe the attack formally, and do a preliminary analysis followed by a detailed description of why the attack works. As before, we assume that all $\binom{m}{3}$ possible 3-combinations of labels are enumerated and let $\Gamma_1, \ldots, \Gamma_{\binom{m}{3}}$ denote these 3-combinations. Thus, each 3-combination is a set of 3 labels.

Attack 2.

Input: r challenge-response pairs with response points $P_1, \ldots, P_r$, respectively, and a threshold τ.

Output: Label(s) with maximum frequency.

1: *Test Set.* Initialize $C \leftarrow \phi$. For $1 \leq i \leq \binom{m}{3}$, if $P_1 \in \mathsf{ch}(\Gamma_i)$, then $C \leftarrow C \cup \{\Gamma_i\}$.
2: *Frequency List.* For each $\Gamma \in C$, initialize $\mathsf{freq}(\Gamma) \leftarrow 1$.
3: **for** $i = 2$ to r **do**
4: For each $\Gamma \in C$, if $P_i \in \mathsf{ch}(\Gamma)$, then $\mathsf{freq}(\Gamma) \leftarrow \mathsf{freq}(\Gamma) + 1$.
5: *Thresholded Subset.* $C^{(\tau)} \leftarrow \{\Gamma \in C | \mathsf{freq}(\Gamma) > \tau\}$.
6: *Frequency of labels.* For each distinct label l in $C^{(\tau)}$ compute:

$$\mathsf{freq}(l) \leftarrow \sum_{\Gamma \in C^{(\tau)} | l \in \Gamma} \mathsf{freq}(\Gamma)$$

7: Output all labels l' such that $\mathsf{freq}(l') = \max_{l \in C^{(\tau)}} \{\mathsf{freq}(l)\}$.

□

The time complexity of the above attack is $O(\binom{m}{3})$ or $O(m^3)$. The space complexity is $O(\gamma\binom{m}{3})$. Continuing with our theoretical treatment in the previous section, we would like to first analyze the expected sizes of the frequency lists before we detail the simulation results of Attack 2 and the reasons for its high success probability.

Let $F^{(i)} = \sum_{\Gamma \in C} \mathsf{freq}(\Gamma)$, denote the cumulative frequency after the ith challenge-response pair. We have seen earlier that:

$$E[F^{(1)}] = E[|C|] = \gamma_1 \binom{m}{3}$$

That is, the expected size of the Test Set is as above. Also, let

$$L^{(i)} = \sum_{\Gamma \in C | l \in \Gamma} \mathsf{freq}(\Gamma)$$

denote the frequency of a label after the ith challenge-response pair. For $L^{(1)}$, we see that each label can occur with $\binom{m-1}{2}$ combinations of the remaining labels. Assuming all these combinations to be uniformly distributed, each combination will have a probability γ_1 of being in C. Thus,

$$E[L^{(1)}] = \gamma_1 \binom{m-1}{2}$$

The above two results can also be obtained differently. Consider the indicator random variable $Y_{i,j}$ which is 1 if $P_i \in \mathsf{ch}(\Gamma_j)$. Also, let $X_{i,j}$ be the indicator random variable which is 1 if Γ_j exists in challenge i (Since $m \leq n$, the jth combination might not even exist in challenge i). Then, we can see that:

$$\begin{aligned} E\left[Y_{1,j}\right] &= \Pr\left[Y_{1,j}=1\right] = \Pr\left[Y_{1,j}=1 | X_{1,j}=1\right] \Pr\left[X_{1,j}=1\right] \\ &\quad + \Pr\left[Y_{1,j}=1 | X_{1,j}=0\right] \Pr\left[X_{1,j}=0\right] \\ &= \Pr\left[Y_{1,j}=1 | X_{1,j}=1\right] \frac{m}{n} \frac{m-1}{n-1} \frac{m-2}{n-2} = \gamma_1 \frac{m}{n} \frac{m-1}{n-1} \frac{m-2}{n-2} \\ &= \gamma_1 \frac{\binom{m}{3}}{\binom{n}{3}} \end{aligned}$$

The above result is true since we are assuming that m is chosen according to Lemma 1. From this, it follows that:

$$E[F^{(1)}] = E[\sum_{j=1}^{\binom{n}{3}} Y_{i,1}] = \sum_{j=1}^{\binom{n}{3}} E\left[Y_{i,1}\right] = \binom{n}{3} \gamma_1 \frac{\binom{m}{3}}{\binom{n}{3}} = \gamma_1 \binom{m}{3}$$

Which is the same as the result obtained above. Now, since each combination contains 3 labels and we assume all the labels to be uniformly distributed over the combinations, we get that:

$$E[L^{(1)}] = \frac{3}{m} \gamma_1 \binom{m}{3} = \gamma_1 \binom{m-1}{2}$$

We now attempt to find the expected number of Γ's in C such that $P_i \in \mathsf{ch}(\Gamma)$, when $i > 1$. We have:

$$\begin{aligned} E[\sum_{j=1}^{\binom{n}{3}} Y_{1,j} Y_{i,j}] &= \sum_{j=1}^{\binom{n}{3}} E\left[Y_{1,j}\right] E\left[Y_{i,j}\right] \\ &= \binom{n}{3} \gamma_1 \gamma_i \frac{\binom{m}{3}^2}{\binom{n}{3}^2} = \gamma_1 \gamma_i \frac{\binom{m}{3}^2}{\binom{n}{3}} \end{aligned}$$

So, after r challenge-response pairs, we have that:

$$\begin{aligned} E[F^{(r)}] &= \gamma_1 \binom{m}{3} + \gamma_1 \gamma_2 \frac{\binom{m}{3}^2}{\binom{n}{3}} + \cdots + \gamma_1 \gamma_r \frac{\binom{m}{3}^2}{\binom{n}{3}} \\ &= \gamma_1 \binom{m}{3} \left(1 + \frac{\binom{m}{3}}{\binom{n}{3}} \sum_{i=2}^{r} \gamma_i\right) \end{aligned}$$

Table 3. Expected values of the number of combinations and labels against actual average.

Simulation	Number of labels		Number of combinations		Secrets
	Actual	Theoretical	Actual	Theoretical	
1	2815.20	2843.80	84457.00	85315.00	3245.00
2	2101.30	2047.80	63040.00	61434.00	2646.00
3	1156.20	1076.50	34687.00	32295.00	2705.30
4	1073.40	1028.90	32201.00	30867.00	2279.00
5	1327.20	1273.70	39816.00	38210.00	2845.70
6	2466.20	2525.00	73985.00	75751.00	2801.00
7	1885.20	1893.10	56555.00	56794.00	3241.70
8	1892.10	1934.70	56764.00	58042.00	2900.70
9	917.57	930.42	27527.00	27913.00	2147.70
10	2539.20	2534.80	76175.00	76044.00	3483.70
Average	1817.36	1808.87	54520.70	54266.50	2829.58

And

$$E[L^{(r)}] = \frac{3}{m} E[F^{(r)}] = \frac{3}{m} \gamma_1 \binom{m}{3} \left(1 + \frac{\binom{m}{3}}{\binom{n}{3}} \sum_{i=2}^{r} \gamma_i \right)$$

Thus, if we run the attack on a set of r challenge-response pairs, we would expect the number of times each combination and each label to appear according to the above equations. We ran a simulation to compare the theoretical expected values against actual mean values. The simulation was run with the following parameters: $n = 112$, $m = 90$, $k = 5$ and $r = 21$. Table 3 shows the results. As can be seen, the theoretical values match well with the experimental results. For each challenge-response pair i, let S_i denote the set of 3 secret labels used by the user to form a convex hull. The last column in the table shows the average of the frequency of occurence of each label in S_1 (corresponding to the Test Set), after 21 challenge-response pairs. Notice that this value is always higher than the average for all labels. By the pigeonhole principle, this means that at least one of the secret labels occurs with a frequency higher than the expected frequency for all labels. This is the motivation behind Attack 2. Since, at least one of the secret labels appears with a frequency higher than the average, there is a non-trivial chance that it will occur with the highest frequency as the output of Attack 2. We give the simulation results for Attack 2 next, following which we attempt to explain why at least one of the secret labels occurs with an above-average frequency.

6.2 Simulation Results for Attack 2

The simulation results for Attack 2 are shown in Table 4. The column labeled "pairs" shows the number of challenge-response pairs used. The column labeled "Secret Appeared" shows the number of times one of the secret labels is the

output of Attack 2, in 100 runs. Thus, this corresponds to the probability of success of Attack 2. As can be seen, with a non-trivial probability at least one of the secret labels appears with the highest frequency, i.e., the output of Attack 2. The value of τ, or the threshold, is chosen such that the size of $C^{(\tau)}$ is at least 50. There is no particular reason for this choice of τ, except to ensure that the size of $C^{(\tau)}$ is reasonably large. As τ is dynamic over different runs, only its average value is shown.

In all the simulation runs the bounding rectangle had end coordinates:

$$(0,0),(13,0),(0,13),(13,13)$$

We used Turk's method to compute a random point within a triangle [23]. Our simulation results suggest that increasing k makes the probability of success lower. However, the probability is still higher than k/n, the success probability of random guess, which is 0.0446 when $n = 112$ and $k = 5$, and 0.075 when $n = 160$ and $k = 12$. It should be noted that increasing k does not increase the time and space complexity of Attack 2, which is always $O(\binom{m}{3})$, although it does affect the probability of success. Our experimental results also indicate that the success probability increases with more challenge-response pairs. Thus, the probability of success of Attack 2 is a function of n, m, k and r.

Table 4. Output of Attack 2

Simulation Number	n	m	k	pairs	Secret Appeared	Average Threshold
1	112	90	5	20	$64/100 = 0.64$	6.4
2				30	$76/100 = 0.76$	7.8
3				50	$88/100 = 0.88$	10.9
4	160	100	12	20	$35/100 = 0.35$	4.8
5				30	$40/100 = 0.40$	5.6
6				50	$48/100 = 0.48$	7.2

6.3 Why does Attack 2 Work

In this section, we give a qualitative explanation for the success of Attack 2. That is, we explain why one of the secret label appearing in C has the highest frequency with high probability. We show this in two steps. First, we show that relative to any point P clicked by the user, there are regions in the rectangle where labels of lattice points have low and high frequencies. Secondly, we reason that the secret labels have a higher probability of being in the high frequency region as compared to non-secret labels.

We assume that the rectangle has coordinates $(0,0),(a,0),(0,b)$ and (a,b) for some positive integers a and b. For simplicity, we assume that $(a+1)(b+1) = m$. That is, the number of possible lattice points that can be contained in the

rectangle is exactly m. Assume that we are given a response point $P \in \mathbb{R}^2$. Let $C = \{\Gamma_i | P \in \mathsf{ch}(\Gamma_i), 1 \leq i \leq \binom{m}{3}\}$. Also, for a label l, define:

$$\mathsf{freq}(l) = \sum_{\Gamma \in C | l \in \Gamma} \mathsf{freq}(\Gamma)$$

For a lattice point I in the rectangle, let $\mathsf{lab}(I)$ denote the label of I. Of interest is the question that which region of the rectangle, relative to P, contains the lattice points whose labels have higher values of $\mathsf{freq}(.)$.

Abusing notation, we shall denote $\mathsf{freq}(\mathsf{lab}(I))$ by $\mathsf{freq}(I)$, when considering a generic label. We can see that $\mathsf{freq}(I) \leq \binom{m-1}{2}$. And it is not hard to see that if $I = P$, then $\mathsf{freq}(I) = \binom{m-1}{2}$. We now consider the case when $P \neq I$. Consider the line segment $\overline{IP}$. Extend this line segment in both directions such that it intersects the boundaries of the rectangle at points R_1 and R_2 as shown in Figure 4. If any lattice point lies on the line segment $\overline{PR_2}$, then all its 3-combinations with I will be in C. We next consider the case when no lattice point except I, lies on the line $\overline{R_1R_2}$. $\overline{R_1R_2}$ thus divides the rectangle into 2 partitions. Denote the set of lattice points in these two partitions by Π_1 and Π_2. Thus, $|\Pi_1| + |\Pi_2| = m - 1$. Now consider two lattice points I_1, I_2 both different from I. A necessary condition for the triangle $\Delta I I_1 I_2$ to contain the point P, is for I_1 and I_2 to be in different partitions[6]. Thus, in this case, $\mathsf{freq}(I) \leq |\Pi_1||\Pi_2|$. We wish to find when this product produces the maximum value. We know that $|\Pi_1| = m - 1 - |\Pi_2|$. Thus,

$$|\Pi_1||\Pi_2| = (m - 1 - |\Pi_2|)|\Pi_2|$$

Differentiating the right hand side with respect to $|\Pi_2|$ and equating it to 0, we see that the above product has a maximum value when $|\Pi_2| = (m-1)/2$. This implies that $|\Pi_1| = (m-1)/2$. Thus, the product above will be maximised if the 2 partitions are *equal*.

However, not all the pairs in $\Pi_1 \times \Pi_2$ will form a convex hull with I containing P. Consider the points $I_1 \in \Pi_1$ and $I_2 \in \Pi_2$. The triangle $\Delta I I_1 I_2$ will contain the point P, if the side $\overline{I_1I_2}$ intersects the segment $\overline{PR_2}$. Similarly, if $\overline{I_1I_2}$ intersects $\overline{R_1I}$ or $\overline{IP}$ (except at the point P), then the corresponding triangle with I does not contain the point P. Thus, the longer the segment $\overline{PR_2}$, the higher will be the number of pairs from $\Pi_1 \times \Pi_2$ intersecting it. This gives us the following result:

$\mathsf{freq}(I)$ *will be maximised if: (1) the line* $\overline{R_1R_2}$ *divides the rectangle into two partitions with an almost equal number of lattice points, (2) the length of the line segment* $\overline{PR_2}$ *is close to the length of* $\overline{R_1R_2}$.

These two observations give us the following informal result:

[6] For suppose that is not the case and both I_1 and I_2 are in Π_1, then all the sides of the triangle $\Delta I I_1 I_2$ never intersect the line segment $\overline{IP}$, except at point I. Hence, the point P cannot be contained in the triangle unless $I = P$. But we have already assumed that not to be true.

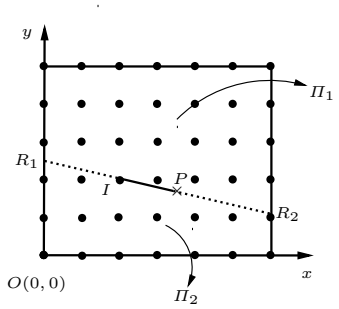

Fig. 4. The 2 partitions Π_1 and Π_2.

Result 1 *Let $P \in \mathbb{R}^2$. Draw a line $\overline{R_1R_2}$ that intersects P and divides the rectangular lattice area into 2 partitions such that the two contain an almost equal number of lattice points. Suppose $\overline{R_1P}$ is shorter than $\overline{R_2P}$. Then the labels of the lattice points around the vicinity of $\overline{R_1P}$ will have higher values of* freq(.). *Furthermore, the labels of the lattice points around the vicinity of $\overline{R_2P}$ will have lower values of* freq(.).

The terms "almost" and "vicinity" used in the above result hold their natural meanings and we do not attempt to rigorously define them. We call the region around the shorter line segment, $\overline{R_1P}$, the high frequency region, and the region around the longer line segment $\overline{PR_2}$, the low frequency region. Sandwitched between these two will be the region containing the lattice points whose labels have mid-range values of freq(.). Figure 5 illustrates this result. We insist that the boundaries of these regions are fuzzy. This analysis is correct except for some degenerate cases; such as when P is at the center of the rectangle. This is not covered by the above result, because $\overline{R_1P} < \overline{R_2P}$ is a necessary condition, which does not hold if P is at the center of the rectangle. In this case, there is an infinite number of ways to partition the rectangle into equal areas. However, apart from these exceptions, we expect the behavior to be similar most of the time. The following theorem proves that given any point P, not at the center of the rectangle, there is always a unique way to partition the rectangle into two equal areas by a line through P and the center of the rectangle.

Theorem 1. *Let R be a rectangle in the xy-plane of real numbers, with vertices $(0,0)$, $(a,0)$, $(0,b)$ and (a,b). Let $C(a/2, b/2)$ be the center of the rectangle R. Let $P \in \mathbb{R}^2$ be a point contained in this rectangle with coordinates (x_P, y_P). Suppose $P \neq C$. Then, the line $\overline{PC}$ is the unique line that divides R into two polygons of equal areas.*

Proof. Any line through C divides the rectangle into equal areas. Therefore, given a point P, the line $\overline{PC}$ will divide the rectangle into equal areas. Now, suppose there is another line L, that goes through P and not through C and divides the rectangle into two polygons of equal areas. Consider the line L' that

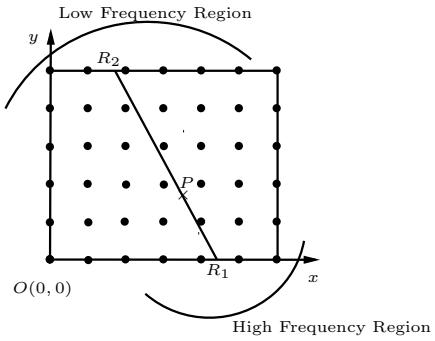

Fig. 5. The high and low bandwidth regions.

is parallel to L and goes through point C. Thus L' also divides the rectangle into two polygons of equal areas. However, L' is different from L since it does not go through P. But this implies that one of the two polygons resulting from L is contained in one of the polygons resulting from L'. This means that the area of the polygon is smaller than half. A contradiction. Therefore, the line $\overline{PC}$ is the unique line dividing a rectangle into equal areas.

The equation of the line $\overline{PC}$ is given by:

$$\begin{aligned}
&\frac{y - y_P}{x - x_P} = \frac{b/2 - y_P}{a/2 - x_P} \\
\Rightarrow\ &(a/2 - x_P)(y - y_P) = (x - x_P)(b/2 - y_P) \\
\Rightarrow\ &(a - 2x_P)(y - y_P) = (x - x_P)(b - 2y_P) \\
\Rightarrow\ &ay - ay_P - 2x_P y + 2x_P y_P = bx - 2xy_P - bx_P + 2x_P y_P \\
\Rightarrow\ &(a - 2x_P)y = (b - 2y_P)x + ay_P - bx_P
\end{aligned}$$

which holds if $x_P \neq a/2$. If $x_P = a/2$, then the equation of the line $\overline{PC}$ is $y = a/2$. □

Theorem 1 allows us to construct the line partitioning the rectangle into two equal parts through the point P. Figure 6 shows the line through the point P during a simulation run. The parameters used were $a = b = 9$, $n = 125$, $m = 100$ and $k = 5$. The triangle shown is the convex hull of the 3 secret labels chosen at random by the user (simulated). As the figure illustrates, the lattice points with the highest values of $\mathsf{freq}(.)$ are populated around the shorter line segment $\overline{R_1 P}$ and the lattice points with the lowest values of $\mathsf{freq}(.)$ are populated around the longer line segment $\overline{R_2 P}$. Incidentally, the figure also shows one of the secret labels (label not shown) appearing in the high frequency region. When $m < (a+1)(b+1)$, there are some "holes" in the rectangle, however the above results still give us a good approximation.

Equipped with this knowledge, we can finally give a reason for the success of Attack 2. Let $S \in C$ be the 3-combination selected by the user to form the

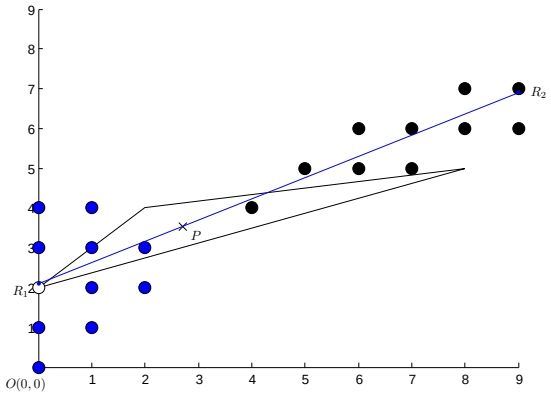

Fig. 6. A simulation run showing the low and high frequency regions.

convex hull containing P. Let the 3 secret labels in S be l_1, l_2 and l_3. Since $\mathsf{ch}(S)$ is a triangle, at least one of l_1, l_2 and l_3 will be in the high frequency region with high probability. To see why this is true, we see that the only way for this not to be true is for one element of S to be in the low frequency region and the other two in the average frequency region. But compared to all possible triangles, the number of such triangles is small. Thus with high probability, at least one of the secrets will be in the high frequency region. Suppose that point is l_1. This means that $\mathsf{freq}(l_1)$ will have a high value relative to most labels. This in turn means that l_1 will be in a high number of 3-combinations in C. Since, the number of such 3-combinations is high, given r challenge-response pairs, these 3-combinations will have a higher value of $\mathsf{freq}(.)$ with high probability, which implies that the frequency of l_1 will be high. It should be noted that l_1 may not appear in some challenges or it could be in the low frequency region in some challenges (because one of the other two secret labels is in the high frequency region). However, on average, a secret label always has a higher probability to be in the high frequency region than a non-secret label. This makes the most frequent label a secret label with high probability. This explains why Attack 2 is successful with high probability.

Improved Variant of Attack 2. The above analysis gives us an interesting way to improve Attack 2. Given r challenge-response pairs, we choose the pair as the Test Set, which has P closest to the boundary of the rectangle. This will mean that with high probability, one of the secret labels will be near the edge and hence will have a high value of $\mathsf{freq}(.)$. Our test results show that indeed this increases the success probability of the attack. The results are shown in Table 5. See in contrast the results obtained in Table 4. We call this variant of Attack 2, the Chosen Test Set Attack.

Table 5. Output of the Chosen Test Set Attack

Simulation Number	n	m	k	pairs	Secret Appeared	Sessions
1	112	90	5	20	$77/100 = 0.77$	10
2				30	$83/100 = 0.83$	14
3				50	$95/100 = 0.95$	20
4	160	100	12	20	$50/100 = 0.50$	77
5				30	$67/100 = 0.67$	86
6				50	$78/100 = 0.78$	123
7	320	200	12	20	$46/100 = 0.46$	83
8				30	$46/100 = 0.46$	125
9				50	$59/100 = 0.59$	163
10	357	200	25	20	$35/100 = 0.35$	330

6.4 Impersonation using Attack 2

Attack 2 (and its variant), outputs one of the secret labels with a non-negligible probability, say $p(n, m, k, r)$ or $p(m, k, r)$, since n is dependent on m and k. We abbreviate this probability as p. One can run the attack multiple times to obtain the whole set of secrets. But that requires in the order of kr challenge-response pairs. While this number is not huge, the adversary can still impersonate $\mathcal{H}$ with high probability even after observing fewer challenge-response pairs. We see that $\mathcal{A}$ does not need to find all the k secrets in order to impersonate $\mathcal{H}$. The impersonation process is described below:

Impersonate $\mathcal{H}$.
Input: t sets of r challenge-response pairs.
Output: 1 if successful, 0 if unsuccessful.

1: *Obtain Secrets.* Run Chosen Test Set Attack on each set of r challenge-response pairs, to obtain the set of labels $L = \{l_1, l_2, \ldots, l_t\}$.
2: *Impersonate $\mathcal{H}$.* Initiate an identification session with $\mathcal{C}$.
3: **for** each of the r_0 challenges sent by $\mathcal{C}$ **do**
4: **if** only one label from L is in the challenge **then**
5: Click on the lattice point of that label.
6: **else if** two labels from L are in the challenge **then**
7: Randomly click any point on the line connecting the two corresponding lattice points.
8: **else if** three or more labels from L are in the challenge **then**
9: Click a random point contained in the convex hull of the corresponding lattice points.
10: **else if** no label from L is in the challenge **then**
11: Click a random point contained in the rectangle.
12: If $\mathcal{C}$ outputs `accept`, then output 1, else output 0.

□

The probability that "Impersonate $\mathcal{H}$" outputs 1, depends in part on the success probability of the Chosen Test Set Attack. This also suggests that once $k-2$ secret labels are obtained, they are enough to impersonate $\mathcal{H}$ with probability 1. This is true since every challenge will contain at least one of the $k-2$ secret labels, and then the above impersonation process can be used to impersonate $\mathcal{H}$[7]. Thus, if $k=5$, only 3 secret labels are required, and if $k=12$, only 10 secret labels are enough. Thus the effective security of the protocol is $k-2$ secret labels.

Even if the number of secret labels obtained is less than $k-2$, impersonation can still be successful with high probability. For instance, the probability that the t labels in L are all distinct secret labels and the adversary is successful in impersonating $\mathcal{H}$ is:

$$\begin{aligned}
&\frac{k-1}{k}\frac{k-2}{k}\cdots\frac{k-t+1}{k}\left(\Pr[|L|=0]\cdot\frac{1}{2}+(1-\Pr[|L|=0])\cdot 1\right)^{r_0}p^t \\
&=\frac{p^t}{k^{t-1}}(k-1)\cdots(k-t+1)\left(1-\frac{1}{2}\Pr[|L|=0]\right)^{r_0} \\
&=\frac{p^t}{k^{t-1}}(k-1)\cdots(k-t+1)\left(1-\frac{1}{2}(1-(m/n))^t\right)^{r_0} \\
&=\left(\frac{p}{k}\right)^t\frac{k!}{(k-t)!}\left(1-\frac{1}{2}(1-(m/n))^t\right)^{r_0} \qquad (3)
\end{aligned}$$

when $t<k-2$ and:

$$\left(\frac{p}{k}\right)^t\frac{k!}{(k-t)!}$$

when $t\geq k-2$. Here, we have assumed that the probability of success of a random click is 1/2. In actual, it can be considerably less than 1/2. But that does not result in any substantial change in the overall success probability of impersonation. Let us consider the parameter values $k=5$, $n=112$ and $m=90$, and assume that $r_0=10$ and $r=30$. From Table 5, we get the approximate probability of success of the Chosen Test Set Attack as $p=0.95$. For these values, the above probability has the peak value of 0.59 at $t=2$. This implies that even after observing only $tr/r_0=6$ identification sessions, the adversary has a 60 percent chance of getting $t=2$ distinct secret labels and successfully impersonating $\mathcal{H}$. For $k=12$, $n=160$, $m=100$ and $r=50$, we get the approximate value of $p=0.78$ from Table 5. These values give a peak probability value of 0.27 at $t=3$. This means only 15 observed identification sessions. Similarly, for $n=320$, $m=200$, $k=12$ and $r=50$, we get the peak probability 0.15 at $t=2$. These probabilities are non-negligible. Notice that the probability of success through random clicks is less than $(1/2)^{10}\approx 0.00098$.

The output of t trials of Attack 2 can be modeled as following a binomial distribution, where the probability that a secret label is the output is p. Since

[7] There is a small probability that the attacker may fail, due to an inaccurate click. A human user cannot always exactly click the center of an icon or on a line, which may render the clicked point out of the convex hull.

each secret label is equally likely to be the output, we can assume that the probability of each one being the output is p/k. Let X denote the number of trials required before $k-2$ distinct labels are obtained, given that each trial is a success. Then [24, §7.2, p. 334]:

$$E[X] = 1 + \frac{k}{k-1} + \frac{k}{k-2} + \cdots + \frac{k}{3}$$

And since the trials are distributed binomially, we get:

$$\begin{aligned} tp &= 1 + \frac{k}{k-1} + \frac{k}{k-2} + \cdots + \frac{k}{3} \\ \Rightarrow t &= \frac{1}{p}\left(1 + \frac{k}{k-1} + \frac{k}{k-2} + \cdots + \frac{k}{3}\right) \end{aligned}$$

Thus, the expression above gives the expected number of trials of Attack 2 required to get $k-2$ distinct labels. Each trial takes r challenge-response pairs and there are r_0 challenge-response pairs in each identification session. Thus, under Attack 2, the protocol can only be used for rt/r_0 sessions before the adversary has $k-2$ secret labels to impersonate $\mathcal{H}$ with probability 1. The last column in Table 5, under the heading "Sessions", shows the values of rt/r_0 for the corresponding parameters, where t is obtained from the expression above. These values should be seen with caution, as they only give a rough estimate of the number of sessions a particular secret can be used under the attacks mentioned in this paper. We again stress that the adversary can still impersonate $\mathcal{H}$ with a non-negligible probability even with fewer sessions.

The weakness exploited in Attack 2 seems to be an inherent problem for convex hull based protocols. Even if the user is asked to form a convex hull of more than 3 secret icons, the attack can still be applied. This is true since the point clicked by the user will be contained in at least one of the possible 3-combinations of the secret icons.

7 Discussion and Future Work

A shortcoming of this work is the lack of an explicit expression for $p(m, k, r)$, i.e., the success probability of Attack 2 (or its variant). Unfortunately, there does not seem to be a straightforward way of obtaining such an expression. Still, the numerical values of $p(m, k, r)$ obtained indicate that the protocol is insecure even for system parameter values recommended by Wiedenbeck et al. for high security, such as $n = 320$, $m = 200$ and $k = 12$. For example, there is approximately a 15 percent chance that the adversary can impersonate a user after observing only 10 identification sessions with these parameter values. By observing more sessions, the probability can be improved. It is not clear whether Attack 2 can be modified to obtain secrets with a number of challenge-response pairs less than kr. So far, we only know how to obtain the complete set of secret icons by a sequential application of Attack 2.

To mitigate impersonation using Attack 2, we can increase r_0. For instance, for the aforementioned values and $r_0 = 20$, the peak success probability of impersonation is 0.09. However, increasing r_0 increases the number of challenge-response pairs per session. This in turn implies that the number of identification sessions observed, before the adversary can obtain the secrets, decreases. The usability of the system also decreases with increasing r_0. Another way is to increase k. This does not necessarily imply an increase in identification time, since the user has to form a convex hull of only 3 secret labels. The last row of Table 5 shows the probability of success of the Chosen Test Set attack, when $k = 25$ and $m = 200$. While, the success probability is still non-negligible, with these parameter values, Equation 3 indicates that the user can be authenticated for about 330 sessions before secret renewal based on the attacks mentioned in this paper. However, increasing k raises some usability issues. First, remembering 25 graphical icons might not be easy for humans. Secondly, with $k = 25$, an averagely larger number of secret icons are to be displayed on the screen. This means that the convex hull of these icons can occupy a large area of the screen. This makes the probability of success of the random click attack higher.

The reader is encouraged to find new or improved attacks and/or find fixes to the convex hull based human identification protocols. Another interesting future line of research is to find new geometric problems for human identification protocols. Some other existing examples of human identification protocols based on geometric problems appear in [1, 25], but the exact security of these remains unexplored.

8 Conclusion

The Convex Hull Click (CHC) graphical human identification protocol is an interesting alternative to other proposed protocols in literature. The scheme is easy to execute for humans and is apparently more secure as compared to some of the previous approaches. The security of the underlying problem has not been extensively analysed previously. This is partly due to the complex structure of the problem. This work is the first attempt to extensively analyse the protocol. We have shown two attacks on the CHC protocol. The first attack outputs the secret icons with high probability after observing a few authentication sessions. We have proposed a formula which allows to find values of system parameters for which this attack can be avoided. The second attack outputs a secret icon with high probability after observing only a handful of identification sessions. The attack can be improved and then can be used to impersonate the user with a non-trivial probability. Our approach has been as mathematically rigorous as possible. However, the problem is not easy to tackle analytically and computer simulations were needed to supplement the theoretical work. While in its current form, the protocol does seem to have significant weaknesses, research can be done to find some variants of the protocol that are easy for humans to compute while being secure at the same time.

Acknowledgements. We thank Jean-Camille Birget for useful comments and suggestions for improvement on an earlier draft of the paper. Hassan Jameel Asghar was supported by Macquarie University Research Excellence Scholarship (MQRES). 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). Josef Pieprzyk was supported by the Australian Research Council under Grant DP0987734. The work of Huaxiong Wang was supported in part by the Singapore National Research Foundation under Research Grant NRF-CRP2-2007-03 and the Singapore Ministry of Education under Research Grant T206B2204.

References

[1] Leonardo Sobrado and Jean-Camille Birget. Graphical passwords. *The Rutgers Scholar*, 4, 2002. http://rutgersscholar.rutgers.edu/volume04/sobrbirg/sobrbirg.htm.

[2] Huanyu Zhao and Xiaolin Li. S3PAS: A scalable shoulder-surfing resistant textual-graphical password authentication scheme. In *Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'2007)*, pages 467–472. IEEE Computer Society, 2007.

[3] Susan Wiedenbeck, Jim Waters, Leonardo Sobrado, and Jean-Camille Birget. Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI'2006)*, pages 177–184. ACM, 2006.

[4] Tsutomu Matsumoto and Hideki Imai. Human identification through insecure channel. In *Advances in Cryptology – EUROCRYPT'91*, volume 547 of *Lecture Notes in Computer Science*, pages 409–421, Berlin, 1991. Springer-Verlag.

[5] Chih-Hung Wang, Tzonelih Hwang, and Jiun-Jang Tsai. On the Matsumoto and Imai's human identification scheme. In *Advances in Cryptology – EUROCRYPT'95*, volume 921 of *Lecture Notes in Computer Science*, pages 382–392, Berlin / Heidelberg, 1995. Springer.

[6] Tsutomu Matsumoto. Human-computer cryptography: An attempt. In *Proceedings of the 3rd ACM Conference on Computer and Communications Security (CCS'96)*, pages 68–75. ACM, 1996.

[7] Shujun Li and Heung-Yeung Shum. Secure Human-Computer Identification against peeping attacks (SecHCI): A survey. Technical report available online at http://www.hooklee.com/Papers/SecHCI.pdf, 2003.

[8] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 52–66. Springer-Verlag, Berlin, 2001.

[9] Daphna Weinshall. Cognitive authentication schemes safe against spyware (short paper). In *Proceedings of IEEE Symposium on Security and Privacy (SP'2006)*, pages 295–300. IEEE Computer Society, 2006.

[10] Xiaole Bai, Wenjun Gu, S. Chellappan, Xun Wang, Dong Xuan, and Bin Ma. PAS: Predicate-based Authentication Services against powerful passive adversaries. In *Proceedings of Annual Computer Security Applications Conference (ACSAC'2008)*, pages 433–442. IEEE Computer Society, 2008.

[11] Ming Lei, Yang Xiao, Susan V. Vrbsky, and Chung-Chih Li. Virtual password using random linear functions for on-line services, ATM machines, and pervasive computing. *Computer Communications*, 31(18):4367–4375, 2008.

[12] Philippe Golle and David Wagner. Cryptanalysis of a cognitive authentication scheme (extended abstract). In *Proceedings of IEEE Symposium on Security and Privacy (SP'2007)*, pages 66–70. IEEE Computer Society, 2007.

[13] Shujun Li, Hassan Jameel Asghar, Josef Pieprzyk, Ahmad-Reza Sadeghi, Roland Schmitz, and Huaxiong Wang. On the security of PAS (Predicate-based Authentication Service). In *Proceedings of the 2009 Annual Computer Security Applications Conference (ACSAC'2009)*, pages 209–218. IEEE Computer Society, 2009.

[14] Shujun Li, Syed Ali Khayam, Ahmad-Reza Sadeghi, and Roland Schmitz. Breaking randomized linear generation functions based virtual password system. In *Proceedings of IEEE International Conference on Communications (ICC 2010)*. IEEE, 2010.

[15] Xiang-Yang Li and Shang-Hua Teng. Practical human-machine identification over insecure channels. *Journal of Combinatorial Optimization*, 3(4):347–361, 199.

[16] Hassan Jameel, Riaz Shaikh, Heejo Lee, and Sungyoung Lee. Human identification through image evaluation using secret predicates. In *Topics in Cryptology – CT-RSA 2007*, volume 4377 of *Lecture Notes in Computer Science*, pages 67–84, Berlin / Heidelberg, 2007. Springer.

[17] Hassan Jameel, Riaz Shaikh, Le Hung, Yuan Wei, Syed Raazi, Ngo Canh, Sungyoung Lee, Heejo Lee, Yuseung Son, and Miguel Fernandes. Image-feature based human identification protocols on limited display devices. In *Information Security Applications, 9th International Workshop, WISA 2008, Jeju Island, Korea, September 23-25, 2008, Revised Selected Papers*, volume 5379 of *Lecture Notes in Computer Science*, pages 211–224, Berlin / Heidelberg, 2009. Springer.

[18] Hassan Jameel Asghar, Josef Pieprzyk, and Huaxiong Wang. A new human identification protocol and coppersmith's baby-step giant-step algorithm. In Jianying Zhou and Moti Yung, editors, *ACNS*, volume 6123 of *Lecture Notes in Computer Science*, pages 349–366, 2010.

[19] Rachna Dhamija and Adrian Perrig. Déjà Vu: A user study using images for authentication. In *Proceedings of 9th USENIX Security Symposium*, pages 45–58. USENIX Association, 2000.

[20] Volker Roth, Kai Richter, and Rene Freidinger. A PIN-entry method resilient against shoulder surfing. In *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS'2004)*, pages 236–245. ACM, 2004.

[21] Hirokazu Sasamoto, Nicolas Christin, and Eiji Hayashi. Undercover: Authentication usable in front of prying eyes. In *Proceeding of 26th Annual SIGCHI Conference on Human Factors in Computing Systems (CHI'2008)*, pages 183–192. ACM, 2008.

[22] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[23] Greg Turk. *Graphics gems*, pages 24–28. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

[24] Sheldon Ross. *A First Course in Probability*. Prentice Hall, 7th edition, 2006.

[25] Shushuang Man, Dawei Hong, Jean-Camille Birget, and Manton Mathews. A shoulder-surfing resistant graphical password scheme. http://clam.rutgers.edu/ birget/grPssw/manDawei.pdf, 2005.

A Turk's Method of Generating a Random Point Inside a Triangle

Let A, B and C be the vertices of a triangle. Let s and t be uniform random real numbers in the interval $[0, 1]$. Turk's method generates a random point P contained in the triangle as follows [23]:

Turk's Method.
Input: Vertices A, B and C and the random numbers s and t.
Output: Random point P contained in the triangle ΔABC.

1: **if** $s + t > 1$ **then**
2: $\quad s \leftarrow 1 - s$.
3: $\quad t \leftarrow 1 - t$.
4: $a \leftarrow 1 - s - t$.
5: $b \leftarrow s$.
6: $c \leftarrow t$.
7: Output $P \leftarrow aA + bB + cC$.