BehavioCog: An Observation Resistant Authentication Scheme

Jagmohan Chauhan[1,2], Benjamin Zi Hao Zhao[1], Hassan Jameel Asghar[2], Jonathan Chan[2], and Mohamed Ali Kaafar[2]

[1] UNSW, Sydney, Australia
[2] Data61, CSIRO, Sydney, Australia
{jagmohan.chauhan, ben.zhao, hassan.asghar, jonathan.chan, dali.kaafar}@data61.csiro.au

Abstract. We propose that by integrating behavioural biometric gestures—such as drawing figures on a touch screen—with challenge-response based cognitive authentication schemes, we can benefit from the properties of both. On the one hand, we can improve the usability of existing cognitive schemes by significantly reducing the number of challenge-response rounds by (partially) relying on the hardness of mimicking carefully designed behavioural biometric gestures. On the other hand, the observation resistant property of cognitive schemes provides an extra layer of protection for behavioural biometrics; an attacker is unsure if a failed impersonation is due to a biometric failure or a wrong response to the challenge. We design and develop a prototype of such a "hybrid" scheme, named BehavioCog. To provide security close to a 4-digit PIN—one in 10,000 chance to impersonate—we only need two challenge-response rounds, which can be completed in less than 38 seconds on average (as estimated in our user study), with the advantage that unlike PINs or passwords, the scheme is secure under observation.

1 Introduction

In Eurocrypt 1991 [31], Matsumoto and Imai raised an intriguing question: Is it possible to authenticate a user when someone is observing? Clearly, passwords, PINs or graphical patterns are insecure under this threat model. Unfortunately, a secure observation resistant authentication scheme is still an open problem. Most proposed solutions are a form of shared-secret challenge-response authentication protocols relying on human cognitive abilities, henceforth referred to as cognitive schemes. To minimize cognitive load on humans, the size $|R|$ of the response space R needs to be small, typically ranging between 2 and 10 [20, 26, 40, 5]. Since anyone can randomly guess the response to a challenge with probability $|R|^{-1}$, the number of challenges (or rounds) per authentication session needs to be increased, thereby increasing authentication time. For example, to achieve a security equivalent to (guessing) a six digit PIN, i.e., 10^{-6}, the cognitive authentication scheme (CAS) [40] requires 11 rounds resulting in 120 seconds to authenticate, while the Hopper and Blum (HB) scheme [20] requires 20 rounds

and 660 seconds [42] (See Section 3 and Section 7 for a brief description of these schemes.) An authentication time between 10 to 30 seconds per round is perhaps acceptable if we could reduce the number of rounds, since cognitive schemes provide strong security under observation.

Our idea is to leverage gesture-based behavioural biometrics by mapping $|R|$ different gesture-based *symbols* (words or figures) to the $|R|$ different responses. Note that both the mapping and the symbols are public. The user renders symbols on the touch screen of a device, e.g., a smartphone. A classifier decides whether the rendering matches that of the target user. We could tune the classifier to achieve a true positive rate (TPR) close to 1, while giving it some leverage in the false positive rate (FPR), say 0.10. The attacker has to correctly guess the cognitive response and correctly mimic the target user's gesture. We now see how we can reduce the number of rounds of the cognitive scheme. Suppose $|R| = 4$ in the cognitive scheme. If the average FPR of rendering four symbols, (i.e., success rate of mimicking a target user's rendering of the four symbols), is 0.10, then the probability of randomly guessing the response to a challenge can be derived as $\text{FPR} \times |R|^{-1} = 0.10 \times 0.25 = 0.025$. Thus, only 4 rounds instead of 11 will make the guess probability lower than the security of a 6-digit PIN. Reducing the number of rounds enhances the usability of existing cognitive based schemes by minimizing the authentication time and reducing cognitive load on the user. The idea also prevents a possible attack on standalone behavioural biometric based authentication. Standalone here mean schemes which only rely on behavioural based biometrics. Minus the cognitive scheme, an imposter can use the behavioural biometric system as an "oracle" by iteratively adapting its mimicking of the target user's gestures until it succeeds. Integrated with a cognitive scheme, the imposter is unsure whether a failed attempt is due to a biometric error or a cognitive error, or both. The benefit appears mutual.

Combining the two authentication approaches into a "hybrid" scheme is not easy, because: (a) to prevent observation attacks, the behavioural biometric gestures should be hard to mimic. Simple gestures (swipes) are susceptible to mimicry attacks [23], while more complex gestures [32, 34] (free-hand drawings) only tackle shoulder-surfing attacks, and (b) the cognitive schemes proposed in the literature are either not secure [40] against known attacks or not usable due to high cognitive load (see Section 7). This leads to our other main contributions:

- We propose a new gesture based behavioural biometric scheme that employs a set of words constructed from certain letters of English alphabets (e.g., *b*,*f*,*g*,*x*,*m*). Since such letters are harder to write [22], we postulate that they might show more inter-user variation while being harder to mimic. Our results indicate plausibility of this claim; we achieve an average FPR of 0.05 under video based observation attacks.
- We propose a new cognitive authentication scheme inspired from the HB protocol [20] and the Foxtail protocol [26, 1]. The scheme can be thought of as a contrived version of learning with noisy samples, where the noise is partially a function of the challenge. The generalized form of the resulting

scheme is conjectured to resist around $|R| \times n$ challenge-response pairs against computationally efficient attacks; n being the size of the problem.
- We combine the above two into a hybrid authentication scheme called BehavioCog and implement it as an app on Android smartphones. The app is configurable; parameter sizes of both the cognitive (challenge size, secret size, etc.) and behavioural biometric (symbols, amount of training, etc.) components can be tuned at set up.
- We extensively analyze the usability, security and repeatability of our scheme with 41 users. The average authentication time for each round is as low as 19 seconds, and we achieve security comparable to a 4-digit and 6-digit PIN in just 2 and 3 rounds, respectively, even under observation attacks. Our user study assesses security against video-based observation by recording successful authentication sessions and then asking users to impersonate the target users. None of the video based observation attacks were successful (with two rounds in one authentication session). We show that by carefully designing the training module, the error rate in authentication can be as low as 14% even after a gap of one week, which can be further reduced by decreasing the secret size.

We do not claim that our idea completely solves the problem raised by Matsumoto and Imai, but believe it to be a step forward towards that goal, which could potentially revive interest in research on cognitive authentication schemes and their application as a separate factor in multi-factor authentication schemes.

2 Overview of BehavioCog

We begin with defining authentication schemes and the adversarial model, followed by the overview of our BehavioCog scheme.

2.1 Preliminaries

Authentication Schemes: A *shared-secret challenge-response* authentication scheme consists of two protocols: *registration* and *authentication*, between the a user (prover) $\mathcal{U}$, and an authentication service (verifier) $\mathcal{S}$, who share a secret x from a secret space X during registration. The authentication phase is as follows: for γ rounds, $\mathcal{S}$ sends a challenge $c \in C$ to $\mathcal{U}$, who sends the response $r = f(x, c)$ back to $\mathcal{S}$. If all γ responses are correct $\mathcal{S}$ accepts $\mathcal{U}$. Here, C is the challenge space, and r belongs to a response space R. We refer to the function $f : X \times C \to R$ as the cognitive function. It has to be computed mentally by the user. Note that server also computes the response (as the user and the server share the same secret). Apart from the selected secret $x \in X$, everything else is public. A challenge and a response from the same round shall be referred to as a challenge-response pair. An authentication session, consists of γ challenge-response pairs. In practice, we assume $\mathcal{U}$ and $\mathcal{S}$ interact via the $\mathcal{U}$'s device, e.g., a smartphone.

Adversarial Model: We assume a passive adversary $\mathcal{A}$ who can observe one or more authentication sessions between $\mathcal{U}$ and $\mathcal{S}$. The goal of $\mathcal{A}$ is to impersonate $\mathcal{U}$ by initiating a new session with $\mathcal{S}$, either via its own device or via $\mathcal{U}$'s device, and making it accept $\mathcal{A}$ as $\mathcal{U}$. In practice, we assume that $\mathcal{A}$ can observe the screen of the device used by $\mathcal{U}$. This can be done either via shoulder-surfing (simply by looking over $\mathcal{U}$'s shoulder) or via a video recording using a spy camera. The attacker is a human who is given access to the user touch gestures via video recordings and then tries to mimic the user. The attacker can view the video any number of times including pausing, rewinding, forwarding, etc. Note that the original threat model from Matsumoto and Imai also assumes that the device as well as the communication channel between the device and $\mathcal{S}$ are insecure. Our threat model is slightly restricted.

2.2 The BehavioCog Scheme

The main idea of BehavioCog hybrid authentication scheme is as follows. Instead of sending the response r to a challenge c from $\mathcal{S}$, $\mathcal{U}$ renders a *symbol* corresponding to r (on the touch screen of the device), and this rendered symbol is then sent to $\mathcal{S}$. More specifically, we assume a set of symbols denoted Ω, e.g., a set of words in English, where the number of symbols equals the number of responses $|R|$. Each response $r \in R$ is mapped to a symbol in Ω. The symbol corresponding to r shall be represented by $\mathrm{sym}(r)$. Upon receiving the rendering of $\mathrm{sym}(r)$, $\mathcal{S}$ first checks if the rendered symbol "matches" a previously stored rendering from $\mathcal{U}$ (called template) by using a classifier D and then checks if the response r is correct by computing f. If the answer to both is yes in each challenge-response round, $\mathcal{S}$ accepts $\mathcal{U}$.

The scheme consists of setup, registration and authentication protocols. We begin by detailing the cognitive scheme first. Assume a global pool of n objects (object is a generic term and can be instantiated by emojis, images or alphanumerics). We used pass-emojis in the paper. A secret $x \in X$ is a k-element subset of the global pool of objects. Thus, $|X| = \binom{n}{k}$. Each object of x is called a pass-object, and the remaining $n-k$ objects are called decoys. The challenge space C consists of pairs $c = (a, w)$, where a is an l-element sequence of objects from the global pool, and w is an l-element sequence of integers from $\mathbb{Z}_d$, where $d \geq 2$. Members of w shall be called weights. The ith weight in w is denoted w_i and corresponds to the ith element of a, i.e., a_i. The notation $c \in_U C$ means sampling a random l-element sequence of objects a and a random l-element sequence of weights w. The cognitive function f is defined as

$$f(x, c) = \begin{cases} \left(\sum_{i|a_i \in x} w_i\right) \bmod d, & \text{if } x \cap a \neq \emptyset \\ r \in_U \mathbb{Z}_d, & \text{if } x \cap a = \emptyset. \end{cases} \tag{1}$$

That is, sum all the weights of the pass-objects in c and return the answer modulo d. If no pass-object is present then a random element from $\mathbb{Z}_d$. is returned. The notation $\in_U$ means sampling uniformly at random. It follows that the response space $R = \mathbb{Z}_d$ and $|R| = d$. Now, let Ω be a set of d symbols,

e.g., the words `zero`, `one`, `two`, and so on. The mapping sym : $\mathbb{Z}_d \to \Omega$ is the straightforward lexicographic mapping. Note that this mapping is public. We assume a $(d+1)$-classifier D (see Section 4) which when given as input the templates of all symbols in Ω, and a rendering purported to be of some symbol from Ω, outputs the corresponding symbol in Ω if the rendering matches any of the symbol templates. If no match is found, D outputs "none." D needs a certain number of renderings of each symbol to build its templates, which we denote by t (e.g., $t = 3, 5$ or 10).

The setup phase consists of $\mathcal{S}$ publishing the parameters n, k, l and d (e.g., $n = 180$, $k = 14$, $l = 30$, $d = 5$), a pool of n objects (e.g., emojis), a set of d symbols Ω (e.g., words), the map sym from $\mathbb{Z}_d$ to Ω, the (untrained) classifier D, and t Figure 1 describes the registration and authentication protocols. Since the registration protocol is straightforward, we only briefly describe the authentication protocol here. $\mathcal{S}$ initializes an *error* flag to 0 (Step 1). Then, for each of the γ rounds, $\mathcal{S}$ sends $c = (a, w) \in_U C$ to $\mathcal{U}$ (Step 3). $\mathcal{U}$ computes f according to Eq. 1, and obtains the response r (Step 4). $\mathcal{U}$ gets the symbol to be rendered through sym(r), and sends a rendering of the symbol to $\mathcal{S}$ (Step 5). Now, $\mathcal{S}$ runs the trained classifier D on the rendered symbol (Step 6). If the classifier outputs "none," $\mathcal{S}$ sets the error flag to 1 (Step 8). Otherwise, D outputs the symbol corresponding to the rendering. Through the inverse map, $\mathcal{S}$ gets the response r corresponding to the symbol (Step 10). Now, if $x \cap a = \emptyset$, i.e., none of the pass-objects are in the challenge, then any response $r \in \mathbb{Z}_d$ is valid, and therefore $\mathcal{S}$ moves to the next round. Otherwise, if $x \cap a \neq \emptyset$, $\mathcal{S}$ further checks if r is indeed the correct response by computing f (Step 11). If it is incorrect, $\mathcal{S}$ sets the error flag to 1 (Step 12). Otherwise, if the response is correct, $\mathcal{S}$ moves to the next round. If after the end of γ rounds, the error flag is 0, then $\mathcal{S}$ accepts $\mathcal{U}$, otherwise it rejects $\mathcal{U}$ (Step 13).

1: Registration.

1 $\mathcal{U}$ and $\mathcal{S}$ share a secret $x \in X$.
2 For each symbol in Ω, $\mathcal{U}$ sends t renderings to $\mathcal{S}$.
3 For each symbol in Ω, $\mathcal{S}$ trains D on the t renderings to obtain $\mathcal{U}$'s template.
4 The secret consists of x and the d templates.

2: Authentication.

1 $\mathcal{S}$ sets err = 0.
2 **for** γ *rounds* **do**
3 $\quad$ $\mathcal{S}$ samples $c = (a, w) \in_U C$ and sends it to $\mathcal{U}$.
4 $\quad$ $\mathcal{U}$ computes $r = f(x, c)$.
5 $\quad$ $\mathcal{U}$ renders the symbol sym(r), and sends it to $\mathcal{S}$.
6 $\quad$ $\mathcal{S}$ runs D on the rendering.
7 $\quad$ **if** *D outputs "none"* **then**
8 $\quad\quad$ $\mathcal{S}$ sets err = 1.
9 $\quad$ **else**
10 $\quad\quad$ $\mathcal{S}$ obtains r corresponding to the symbol output by D.
11 $\quad\quad$ **if** $x \cap a \neq \emptyset$ *and* $r \neq f(x, c)$ **then**
12 $\quad\quad\quad$ $\mathcal{S}$ sets err = 1.
13 **If** err = 1, $\mathcal{S}$ rejects $\mathcal{U}$; otherwise it accepts $\mathcal{U}$.

Fig. 1: The registration and authentication protocols of BehavioCog.

3 The Cognitive Scheme

Our proposed cognitive scheme can be thought of as an amalgamation of the HB scheme based on the learning parity with noise (LPN) problem [20], and the Foxtail scheme (with window) [26, 1]. Briefly, a round of the HB protocol consists of an n-element (random) challenge from $\mathbb{Z}_2^n$. The user computes the dot product modulo 2 of the challenge with a binary secret vector from $\mathbb{Z}_2^n$. With a predefined probability η, say 0.25, the user flips the response, thus adding noise. When the series of challenge-response pairs are written as a system of linear congruences, solving it is known as the LPN problem. The HB protocol can be generalized to a higher modulus d [20]. The Foxtail scheme consists of dot products modulo 4 of the secret vector with challenge vectors from $\mathbb{Z}_4^n$. If the result of the dot product is in $\{0, 1\}$ the user sends 0 as the response, and 1 otherwise. The "window-based" version of Foxtail, consists of challenges that are of length $l < n$. More specifically, we use the idea of using an l-element challenge from the Foxtail with window scheme. However instead of using the Foxtail function, which maps the sum of integers modulo $d = 4$, to 0 if the sum is in $\{0, 1\}$, and 1 otherwise, we output the sum itself as the answer. The reason for that is to reduce the number of rounds, i.e., γ, for a required security level (the success probability of random guess is $\frac{1}{2}$ in one round of the Foxtail scheme). Now if we allow the user to only output 0 in case none of its pass-objects are present in a challenge, the output of f is skewed towards 0, which makes the scheme susceptible to a statistical attack proposed by Yan et al. [42] outlined in Section 3.1. To prevent such attacks, we ask the user to output a random response from $\mathbb{Z}_d$ (not only zero) in such a case. Due to the random response, we can say that the resulting scheme adds noise to the samples (challenge-response pairs) collected by $\mathcal{A}$, somewhat similar in spirit to HB. The difference is that in our case, the noise is (partially) a function of the challenge, whereas in HB the noise is independently generated with a fixed probability and added to the sum. We remark that if we were to use the HB protocol with a restricted window (i.e., parameter l) and restricted Hamming weight (i.e., parameter k), the resulting scheme is not based on the standard LPN problem. Having laid out the main idea behind the cognitive scheme, we now discuss its security.

3.1 Security Analysis

Due to space limitation we only discuss the general results here and leave their derivation and detailed explanation to Appendix A. This analysis is based on well-known attacks on cognitive authentication schemes. We do not claim this analysis to be comprehensive, as new efficient attacks may be found in the future. Nevertheless, the analysis shown here sheds light on why the scheme was designed the way it is.

Random Guess Attack: The success probability p_{RG} of a random guess is conditioned on the event $a \cap x$ being *empty* or not. Since this event shall be frequently referred to in the text, we give it a special name: the *empty case*. The probability

of the empty case is $\mathbb{P}[|a \cap x| = 0] \doteq p_0 = \binom{n-k}{l}/\binom{n}{l}$. We shall use the notation $\doteq$ when defining a variable. Thus, $p_{\mathrm{RG}} = p_0 + (1-p_0)\frac{1}{d}$.

Brute Force Attack (BF) and Information Theoretic Bound. This attack outputs a unique candidate for the secret after $m \doteq m_{\mathrm{it}} = -\log_2 \binom{n}{k}/\log_2(p_0 + (1-p_0)\frac{1}{d})$ challenge-response pairs have been observed. We call m_{it}, the information theoretic bound on m. The complexity of the brute force attack is $\binom{n}{k}$.

Meet-in-the-Middle Attack (MitM). This attack [20] works by dividing the search space in half by computing $\frac{k}{2}$-sized subsets of X, storing "intermediate" responses in a hash table, and then finding collisions. The time and space complexity of this attack is $\binom{n}{k/2}$. Note that there could be variants of the meet-in-the-middle attack that could trade less space with time. For this analysis, we focus on the version that is most commonly quoted.

Frequency Analysis. Frequency analysis, proposed by Yan et al. [42],[3] could be done either independently or dependent on the response. In response-independent frequency analysis (RIFA), a frequency table of δ-tuples of objects is created, where $1 \leq \delta \leq k$. If a δ-tuple is present in a challenge, its frequency is incremented by 1. After gathering enough challenge-response pairs, the tuples with the highest or lowest frequencies may contain the k secret objects if the challenges are constructed with a skewed distribution. In the response-dependent frequency analysis (RDFA), the frequency table contains frequencies for each possible response in $\mathbb{Z}_d$, and the frequency of a δ-tuple is incremented by 1 in the column corresponding to the response (if present in the challenge). In Appendix A we show that our scheme is immune to both forms of frequency analysis.

Coskun and Herley Attack. Since only l objects are present in each challenge, the number of pass-objects present is also less than k with high probability. Let u denote the average number of bits of x used in responding to a challenge. The Coskun and Herley (CH) attack [14] states that if u is small, then candidates $y \in X, y \neq x$, that are close to x in terms of some distance metric, will output similar responses to x. If we sample a large enough subset from X, then with high probability there is a candidate for x that is a distance ξ from x. We can remove all those candidates whose responses are far away from the observed responses, and then iteratively move closer to x. The running time of the CH attack is at least $|X|/\binom{\log_2|X|}{\xi}$ [14] where $|X| = \binom{n}{k}$, with the trade off that $m \approx \frac{1}{\epsilon^2}$ samples are needed for the attack to output x with high probability [2, 7]. The parameter ϵ is the difference in probabilities that distance $\xi + 1$ and $\xi - 1$ candidates have the same response as x.

Linearization. Linearization works by translating the observed challenge-response pairs into a system of linear equations (or congruences). If this can be done,

[3] We borrow the term frequency analysis from [4].

then Gaussian elimination can be used to uniquely obtain the secret. In Appendix A, we show two different ways our proposed cognitive schemes can be translated into a system of linear equations with dn unknowns. This means that the adversary needs to observe dn challenge-response pairs to obtain a unique solution through Gaussian elimination. Note that if $\mathcal{U}$ were to respond with 0 in the empty case, then we could obtain a linear system of equations after n challenge-response pairs. The introduction of noise expands the number of required challenge-response pairs by a factor of d. Gaussian elimination is by far the most efficient attack on our scheme, and therefore this constitutes a significant gain. We believe the problem of finding a polynomial time algorithm in (k, l, n) which uses $m < dn$ number of samples (say $(d-1)n$ samples) from the function described in Eq. 1 is an interesting open question.

3.2 Example Parameter Sizes

Table 1 (left) shows example list of parameter sizes for the cognitive scheme. These are obtained by fixing $d = 5$ and changing k, l and n such that p_{RG} is approximately 0.25. We suggest $d = 5$ as a balance between reducing the number of rounds required, i.e., γ, and ease of computing f. The column labelled m_{it} is the information theoretic bound to uniquely obtain the secret. Thus, the first two suggestions are only secure with $\leq m_{\text{it}}$ observed samples. The complexity shown for both the meet-in-the-middle attack (MitM) and Coskun and Herley (CH) attack represents time as well as space complexity. The last column is Gaussian elimination (GE), for which the required number of samples is calculated as dn. For other attacks, we show the minimum number of required samples m, such that $m \geq m_{\text{it}}$ and the complexity is as reported. We can think of the last two suggested sizes as secure against an adversary with time/memory resources $\approx 2^{70}/2^{40}$ (medium strength) and $\approx 2^{80}/2^{50}$ (high strength), respectively. The medium and high strength adversaries are defined in terms of the computational resources they possess. In general, there can be many levels of strength (by assigning limits of time/space resources an adversary can have). The strength levels are chosen to illustrate how parameter sizes can be chosen against adversarial resources. The parameter sizes are chosen such that the attack complexity vs the number of samples required are as given in Table 1.

Based on parameter sizes for the cognitive scheme and results from the user study, we recommend the parameters for BehavioCog shown in Table 1 (right). The columns labelled "Sessions" indicate whether the target is a medium-strength or high-strength adversary $\mathcal{A}$. Based on our experiments, CW (complex words) gave the best average FPR of 0.05 (see next section). The "Security" column shows $\mathcal{A}$'s probability in impersonating the user by random guess and mimicking the corresponding behavioural biometric symbol. By setting $p_{\text{RG}} = 0.25$ and multiplying it with FPR, we estimate the total impersonation probability of $\mathcal{A}$. For reference, the same probability for a 4-digit PIN is 1×10^{-4}, and for a 6-digit PIN is 1×10^{-6} (but with no security under observation).

(d, k, l, n)	m_{it}	p_{RG}	BF	MitM	CH	GE
$(5, 5, 24, 60)$	11	0.255	2^{22}	2^{12}	2^{11}	poly(n)
Samples required	-	0	11	11	23	300
$(5, 10, 30, 130)$	24	0.252	2^{48}	2^{28}	2^{33}	poly(n)
Samples required	-	0	24	24	24	650
$(5, 14, 30, 180)$	34	0.256	2^{68}	2^{40}	2^{40}	poly(n)
Samples required	-	0	34	34	94	900
$(5, 18, 30, 225)$	44	0.254	2^{87}	2^{51}	2^{51}	poly(n)
Samples required	-	0	44	44	168	1125

(d, k, l, n)	γ	Sessions (med. $\mathcal{A}$)	Sessions (high $\mathcal{A}$)	Ω	Security
$(5, 5, 24, 60)$	1	10	10	CW	1.3×10^{-2}
$(5, 5, 24, 60)$	2	5	5	CW	1.5×10^{-4}
$(5, 5, 24, 60)$	3	3	3	CW	2×10^{-6}
$(5, 10, 30, 130)$	1	24	24	CW	1.3×10^{-2}
$(5, 10, 30, 130)$	2	12	12	CW	1.5×10^{-4}
$(5, 10, 30, 130)$	3	8	8	CW	2×10^{-6}
$(5, 14, 30, 180)$	1	94	34	CW	1.3×10^{-2}
$(5, 14, 30, 180)$	2	47	17	CW	1.5×10^{-4}
$(5, 14, 30, 180)$	3	31	11	CW	2×10^{-6}
$(5, 18, 30, 225)$	1	511	168	CW	1.3×10^{-2}
$(5, 18, 30, 225)$	2	255	84	CW	1.5×10^{-4}
$(5, 18, 30, 225)$	3	170	56	CW	2×10^{-6}

Table 1: Example parameter sizes for cognitive scheme (left) and BehavioCog (right), where m_{it}: information theoretic bound, p_{RG}: random guess probability, BF: Brute Force, MitM: Meet in the Middle, CH: Coksun and Harley, GE: Gaussian Elimination.

4 The Behavioural Biometric Scheme

Our behavioural biometric authentication scheme is based on touch gestures. We first describe the set of symbols followed by the classifier D and finally the identified features. For each symbol in Ω, TPR of D is the rate when it correctly matches $\mathcal{U}$'s renderings of the symbol to $\mathcal{U}$'s template. FPR of D is the rate when it wrongly decides $\mathcal{A}$'s rendering of the symbol matches $\mathcal{U}$'s template.

4.1 Choice of Symbols

We require that symbols be: (a) rich enough to simulate multiple swipes, (b) hard for $\mathcal{A}$ to mimic even after observation, (c) easily repeatable by $\mathcal{U}$ between successive authentications, and (d) easily distinguishable from each other by D. Accordingly, we chose four different sets of symbols (see Table 2). We tried testing all the four sets of symbols in our first phase of the user study to see which one satisfies all the four aforementioned criteria. We found complex words to be the best symbol set and was used in the implementation of our scheme. Note that words or figures are used for behavioural biometrics part while emojis are used for cognitive scheme.

easy words: These English words for the numbers, and serve as the base case.
complex words: Since the letters *b, f, g, h, k, m, n, q, t, u, w, x, y, z* are more difficult to write cursively than others as they contain more turns [22], we hypothesize that words constructed from them might also show more inter-user variation and be difficult to mimic. Our user study shows positive evidence, as complex words were the most resilient against observation attacks. We constructed five words of length 4 from these 14 letters since users find it hard to render higher length words on touchscreen. As it is difficult to construct meaningful words without vowels, we allowed one vowel in each word.

easy figures: This set contains numbers written in blackboard bold shape. A user can render them by starting at the top left most point and traversing in a down and right manner without lifting the finger. This removes the high variability within user's drawings present in the next set of symbols.
complex figures: These figures were constructed by following some principles (to make them harder to mimic): no dots or taps [24, 13], contain sharp turns and angles [34], the users finger must move in all directions while drawing the symbol.

response	0	1	2	3	4
easy words	`zero`	`one`	`two`	`three`	`four`
complex words	`xman`	`bmwz`	`quak`	`hurt`	`fogy`
easy figures					
complex figures					

Table 2: Mapping of responses ($d = 5$) to symbols.

4.2 Choice of Classifier

We picked dynamic time warping (DTW) [33] because: (a) all chosen symbols exhibit features that are a function of time, (b) it shows high accuracy with a small number of training samples (5-10) [17, 32] (to minimize registration time). Given two time series, DTW finds the *optimal warped* path between the two time series to measure the similarity between them [33]. Assume there is a set Q of features, each of which is a time series. Let $\hat{Q}$ represent the set of templates of the features in Q, which are also time series. Given a test sample of these features (for authentication), also represented Q, the multi-dimensional DTW distance between $\hat{Q}$ and Q is defined as [35]: $\text{DTW}(\hat{Q}, Q) = \sum_{i=1}^{|Q|} \text{DTW}(\hat{q}_i, q_i)$, where $\hat{q}_i \in \hat{Q}$ and $q_i \in Q$, are time series corresponding to feature i.

4.3 Template Creation

For each user-symbol pair (each user drawing a particular symbol) we obtain t sample renderings, resulting in t time series for each feature. Fix each feature, we take one of the t time series at a time, compute its DTW distance with the $t-1$ remaining time series, and sum the distances. The time series with the minimum sum is chosen as the *optimal feature template*. The process is repeated for all features to create the template $\hat{Q}$. We created two sets of optimal templates: (1) $\hat{Q}_{\text{sym}}$ to check if $\mathcal{U}$ produced a valid rendering of a symbol from Ω (only using `x`, `y` coordinates) and (2) $\hat{Q}_{\text{user}}$ to check if the rendering comes from the target user $\mathcal{U}$ or an attacker. Basically, the first template set is used to check if the user rendered a symbol from the set of allowed symbols Ω or some random symbol not

in Ω. After this has been ascertained, it is checked whether the symbol is close to the user's template from the other template set (check behavioural biometrics).

4.4 Classification Decision

Given a set of feature values Q from a sample, the decision is made based on whether $\text{DTW}(\hat{Q}, Q)$ lies below the threshold calculated as $\hbar \doteq \mu + z\sigma$. Here μ is the mean DTW distance between the user's optimal template $\hat{Q}$ and all of the user's t samples in the registration phase [27]. σ is the standard deviation, and $z \geq 0$ is a global parameter that is set according to data collected from all users and remains the same for all users. The thresholds $\hbar_{\text{sym}}$ and $\hbar_{\text{user}}$ correspond to $\hat{Q}_{\text{sym}}$ and $\hat{Q}_{\text{user}}$, respectively. The classification works as follows:

Step 1: If for a given challenge $c = (a, w)$, $x \cap a \neq \emptyset$ (i.e., the non-empty case), $\mathcal{S}$ first gets the target symbol by computing f. Target symbol is the symbol corresponding to the correct response. Then, $\mathcal{S}$ rejects $\mathcal{U}$ if the DTW distance between $\hat{Q}_{\text{sym}}$ and the sample is $> \hbar_{\text{sym}}$. Otherwise, $\mathcal{S}$ moves to Step 2. In the empty case, $\mathcal{S}$ computes the DTW distance between the sample and $\hat{Q}_{\text{sym}}$ for each symbol and picks the symbol which gives the least distance. Next, the distance is compared with $\hbar_{\text{sym}}$ for that symbol, and $\mathcal{S}$ accordingly rejects or goes to Step 2.
Step 2: $\mathcal{S}$ computes the DTW distance between the sample and $\hat{Q}_{\text{user}}$ of the symbol. If the distance is $> \hbar_{\text{user}}$, the user is rejected, otherwise it is accepted.

4.5 Feature Identification and Selection

We identify 19 types of features from the literature [41, 13, 11, 36] and obtain 40 features (Table 3), most of which are self explanatory. Explanation of curvature, slope angle and path angle is described in [36]. Device-interaction features were obtained using the inertial motion sensors: accelerometer and gyroscope of the smartphone. Note that our scheme can be used for any device equipped with a touch screen and inertial motion sensors. We perform a standard z-score normalization on each feature. As an example, Appendix B illustrates the discriminatory power of a single feature (`x`). To select the most distinguishing features from the 40 features for each symbol, we created our own variation of sequential forward feature selection (SFS) [15]. See Algorithm 1 in Appendix C. The algorithm takes as an input a list of features Q_{tot} and a symbol, and outputs a selected list of features Q for that symbol. The algorithm starts with an empty list and iteratively adds one feature at a time by keeping TPR = 1.0 and minimizing the FPR values (calculated based on user-adversary pairs, see Section 5) until all features in Q_{tot} are exhausted. At the end, we are left with multiple candidate subsets for Q from which we pick the one with TPR = 1.0 and the least FPR as the final set of features. The algorithm calls the Get z-List algorithm (Algorithm 2 in Appendix C) as a subroutine (based on a similar procedure from [27]). This algorithm computes the z values that give TPR of 1 and the least FPR for

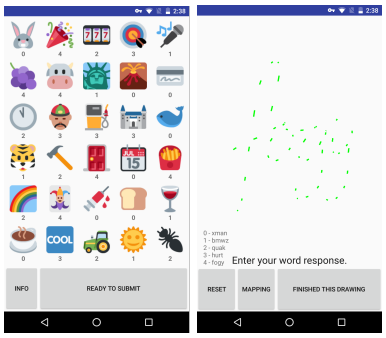
each possible feature subset. The z values give the amount of deviation from the standard deviation.

Table 3: List of features.

Touch feature	Symbol	Stylometric feature	Symbol	Device-interaction feature	Symbol
Coordinates and change in coordinates	$\mathtt{x}$, $\mathtt{y}$, $\delta\mathtt{x}$, $\delta\mathtt{y}$	Top, bottom, left, right most point	TMP, BMP, LMP, RMP	Rotational position of device in space	$\mathtt{R}_x$, $\mathtt{R}_y$, $\mathtt{R}_z$
Velocity along coordinates	$\dot{\mathtt{x}}$, $\dot{\mathtt{y}}$	Width: RMP − LMP, Height: TMP − BMP	width, height	Rate of rotation of device in space	$\mathtt{G}_x$, $\mathtt{G}_y$, $\mathtt{G}_z$
Acceleration along coordinates	$\ddot{\mathtt{x}}$, $\ddot{\mathtt{y}}$	Rectangular area: width × height	area	3D acceleration force due to device's motion and gravity	$\mathtt{A}_x$, $\mathtt{A}_y$, $\mathtt{A}_z$
Pressure and change in pressure	$\mathtt{p}$, $\delta\mathtt{p}$	Width to height ratio	WHR	3D acceleration force solely due to gravity	$\mathtt{g}_x$, $\mathtt{g}_y$, $\mathtt{g}_z$
Size and change in size	$\mathtt{s}$, $\delta\mathtt{s}$	Slope angle	$\theta_{\mathtt{slope}}$	3D acceleration force solely due to device's motion	$\mathtt{a}_x$, $\mathtt{a}_y$, $\mathtt{a}_z$
Force: $\mathtt{p} \times \mathtt{s}$	F	Path angle	$\theta_{\mathtt{path}}$		
Action type: finger lifted up, down or on touchscreen	AT	Curvature	curve		

4.6 Implementation

We implemented BehavioCog for Android smartphones using a set of *twemojis* [38]. We used the parameters $(k, l, n) = (14, 30, 180)$ (corresponding to the medium strength adversary, see Section 3.2). Figure 2 shows an example challenge and response. FastDTW was used to implement DTW [33] with radius 20. The dotted trace in the example response (complex word `fogy`) was a compromise between usability and the difficulty for an attacker to observe fine details.



(a) challenge (b) response

Fig. 2: An example challenge and response in our implementation of BehavioCog.

5 User Study

We did a three phase controlled experimental evaluation of our proposed scheme with 41 participants on a Nexus 5x smartphone after getting the ethics approval.

Phase 1: We collected touch biometric samples from 22 participants: 8 females and 14 males for different symbol sets in two sessions (a week apart) to select the best symbol set (in terms of repeatability and mimicking hardness). As some users contributed samples for multiple symbol sets, we had 40 logical users which were equally divided into four groups, one for each symbol set. Each user did 13 and 3 renderings of each symbol in the first and second session, respectively. The first session was video recorded. Each user acted as an attacker (to mimic a target user's symbol based on video recordings with unrestricted access) for a particular target user and vice versa from the same group.

Phase 2: This phase had a total of 30 participants (11 from Phase 1) and consisted of two sessions (a week apart) to assess the usability and security of BehavioCog. The first session involved cognitive and biometric registration and authentication (video recorded). Second session involved authentication, performing attacks against a target user, and filling a questionnaire. The 30 users were equally divided into three groups: Group 1, 2 and 3 according to the time they spent on registration. All the users chose 14 pass-emojis. 3, 8 and 10 biometric samples for each of the 5 complex words were collected from users in Group 1, Group 2 and Group 3, respectively. The registration for Group 2 and Group 3 users included an extended training game to help them recognize their pass-emojis for better authentication accuracy. The training game was divided into multiple steps in increasing order of difficulty (see Appendix D).Users from Group 3 had to perform double the steps of Group 2 users. Additionally, during Session 2, we asked each user to (a) pick their 14 pass-emojis from the whole pool of emojis, and (b) pick 14 pass-emojis, which they believed belonged to their target (attacked) user.

Phase 3: To find the cause of high number of cognitive errors in Session 2 of Phase 2, we carried out Phase 3 across two sessions (a week apart) with users from Group 3, since they were most familiar with the authentication scheme. First session involved an extended cognitive training: each user was shown 14 pass-emojis one by one for 10 seconds followed by a 3 second cool off period (inspired by cognitive psychology literature [37, 30]), followed by authentication attempts. Session 2 only involved authentication attempts. There are three possible reasons for high cognitive errors: (1) user confuses some of the decoys as pass-emojis since only a subset of pass-emojis are present in a challenge ($l = 30$), (2) user makes errors in computing f, and/or (3) number of pass-emojis is too high (14). To find the exact reason, we asked the user to do the following in order: (a) authenticate six times simply by *selecting* pass-emojis present in the challenge with no weights (to address reason 1); (b) authenticate a further six times, but this time the emojis had weights and the user had to compute f (to

address reason 2), (c) select the 14 pass-emojis from the total pool of 180 (to address reason 3). Phase 3 did not involve any biometrics.

6 Results

Results from Phase 1. We find the best symbol set in terms of repeatability and security by selecting features (through Algorithm 1) for two scenarios: *best case scenario* (secure against random attacks) and *worst case scenario* (secure against video based observation attacks, and repeatability). In both scenarios, first 10 biometric samples from a user (Session 1) are used for training. For the best case, three samples from the same user (Session 1) and three samples from an assigned attacker (Session 1) are used for testing. For the worst case, three samples from the same user (Session 2) and three attacker samples (video based observation attack) are used for testing. Table 4 shows the FPR and top features for each symbol set (TPR is one in all cases). Complex words yield the least FPR which was: 0.0, 0.06, 0.0, 0.2, and 0.0 for `xman`, `bmwz`, `quak`, `hurt` and `fogy`, respectively, in the worst case scenario. All symbol categories have an almost 0% FPR against random attacks. The majority of features providing repeatability and mimicking hardness across all symbol sets are touch and stylometric based. To find out why some symbol sets have poorer average FPR than others in the worst case scenario, we did some further analysis shown in Appendix E.1.

Table 4: Results for best and worst case scenarios for different symbol sets.

Symbol set	Average FPR		Top features	
	best case	worst case	best case	worst case
easy words	0.01	0.24	$\mathtt{x}, \mathtt{y}, \delta\mathtt{x}, \delta\mathtt{y}, \mathtt{TMP}, \theta_{\mathtt{slope}}, \theta_{\mathtt{path}}, \mathtt{R_x}$	$\mathtt{TMP}, \mathtt{height}, \mathtt{WHR}, \theta_{\mathtt{slope}}, \theta_{\mathtt{path}}$
complex words	0.00	0.05	$\mathtt{y}, \delta\mathtt{y}, \mathtt{p}, \mathtt{height}, \mathtt{area}, \theta_{\mathtt{slope}}, \mathtt{R_y}$	$\delta\mathtt{x}, \mathtt{height}, \theta_{\mathtt{path}}$
easy figures	0.01	0.38	$\mathtt{y}, \delta\mathtt{x}, \delta\mathtt{y}, \mathtt{p}, \mathtt{F}, \mathtt{height}, \mathtt{area}, \theta_{\mathtt{slope}}, \theta_{\mathtt{path}}$	$\mathtt{y}, \delta\mathtt{y}, \mathtt{p}, \mathtt{height}$
complex figures	0.01	0.39	$\delta\mathtt{x}$	$\mathtt{x}, \mathtt{TMP}, \mathtt{BMP}$

Results from Phase 2. The goal of Phase 2 was to test the full BehavioCog scheme. We only present selected results related to training and authentication time, errors, attacks and questionnaire here. Detailed results are in Appendix E.

Registration Time: The average time to select 14 pass-emojis was around 2 minutes for all groups. The maximum training time was 12 minutes for Group 3, since it had the most amount of training, and the minimum was 4 minutes for Group 1. High training time is not a major hurdle, because it is a one time process and most of the users reported enjoying the process as it had a "game-like" feel to it (see Appendix E.8). Detailed results are shown in Appendix E.2.

Authentication Time: Table 5 shows the average authentication time (per round) taken by different user groups in the two sessions. Generally, the user spends 15-20 seconds in computing f and 6-8 seconds in entering the biometric response,

which does not change drastically between the two sessions. Group 3 has the least login time (more training results in quicker recognition).

Table 5: Authentication statistics for different user groups.

Group & session	Av. cognitive time (sec)	Av. biometric time (sec)	Av. processing time (sec)	Av. total time (sec)	Success rate (%)	Cognitive errors (%)	Biometric errors (%)
Group 1 - Session 1 (Phase 2)	18.3	7.9	0.7	27.0	38.3	31.6	31.0
Group 2 - Session 1 (Phase 2)	19.8	6.4	0.7	27.0	50.0	18.3	36.0
Group 3 - Session 1 (Phase 2)	12.2	5.6	0.8	18.7	85.0	15.0	0.0
Group 1 - Session 2 (Phase 2)	18.5	7.5	0.7	26.8	26.6	55.0	18.3
Group 2 - Session 2 (Phase 2)	18.4	6.4	0.7	25.6	23.3	55.0	26.6
Group 3 - Session 2 (Phase 2)	15.8	5.4	0.9	22.0	50.0	41.6	8.3
Group 3 - Session 1 (Phase 3)	-	-	-	-	94.0	6.0	-
Group 3 - Session 2 (Phase 3)	-	-	-	-	86.0	14.0	-

Authentication Errors: Table 5 shows the percentage of successful authentication attempts along with the cognitive and biometric errors. There were a total of $v = 60$ authentication attempts (six per user) for each user group in each session. If users were randomly submitting a cognitive response, the probability that i out of v cognitive attempts would succeed is: $p \doteq \binom{v}{i} p_{\text{RG}}^{i} (1 - p_{\text{RG}})^{v-i}$. We consider $i \geq 20$ out of 60 attempts ($< 66\%$ error rate) as statistically significant ($p < 0.05$). Since all groups had cognitive error rate less than 66%, it implies that users were not passing a cognitive challenge by mere chance. Cognitive training aids the user's short term memory, since Group 3 users authenticated successfully 85% of the time, whereas Group 1 users (without cognitive training) were only successful 36% of the time. Group 2 users (with some cognitive training), accrue 18% cognitive errors, similar to Group 3. For Group 2 users most failures originate from biometric errors (they had lesser number of biometric training samples than Group 3). By collecting more biometric data, performance of Group 2 can be made similar to Group 3 with less cognitive training. We see a drastic decrease in the successful authentication attempts in Session 2 from Session 1 especially for Group 3 (from 85% to 50%) and Group 2 (from 50% to 24%). Cognitive errors are predominantly responsible for the drastic decrease as they caused more than half of the authentication attempts to fail for Group 2 and 3, and 40% for Group 1. To find out the actual cause for such a high number of cognitive errors, we did Phase 3 of the study, whose results will be described shortly.

Attack Statistics: We picked those 12 users (9 from Group 3, 2 from Group 2, 1 from Group 1) to be attacked who successfully authenticated 5 out of 6 times in Session 1. Each of the 30 users in the three groups attacked only one of the 12 target users by performing three random and three video based observation attacks totalling 90 attempts. The probability of a random attack can be approximated as $p_{\text{tot}} = p_{\text{RG}} \times \overline{\text{FPR}} \approx 0.256 \times 0.05 \approx 0.013$. Thus i out of $v = 90$ correct guesses would be binomially distributed as $p \doteq \binom{v}{i} p_{\text{tot}}^{i} (1 - p_{\text{tot}})^{v-i}$. We consider $i \geq 4$ as statistically significant ($p < 0.05$). Only 3 attempts (3.33%)

for both attacks were successful, and none of them were consecutive. In all six cases, the target user wrote the words using block letters (easier to mimic [8]).

Questionnaire Results: All 30 users were asked to fill a questionnaire. The results indicate that users find writing the words to be easy on smartphone, users liked playing the training game, and users think that the number of pass-emojis (14) is high. For more details, please see Appendix E.8.

Results from Phase 3. This phase was carried out to find the main cause of cognitive errors and to improve our training to alleviate the issue. The users did 12 authentication attempts each in Sessions 1 and 2. The first 6 involved merely selecting the pass-emojis present whereas the second involved computing f as well. The results are shown in the last two rows of Table 5. The results show that our improved training module (more exposure to each individual pass-emojis followed by blank screens) drastically decreases the error rate. Even after a week's gap the success rate is 86%. We rule out the possibility that the errors in Phase 2 were due to the size of the secret, as the average number of pass-emojis recognized by the users in Sessions 1 and 2 were 13.6 and 13.5, respectively. We also counted the total number of errors made by the users in the first 6 authentication attempts, which turned up 13, and the last 6 authentication attempts, which turned up 11, adding results from both sessions. This shows no evidence that computing f was causing errors. We, therefore, believe that the main cause of errors is due to the user confusing decoy emojis as its pass-emojis since only a subset of the k emojis are present in the challenge (due to l).

7 Related Work

We proposed a new cognitive scheme in our work because existing schemes did not possess all the attributes we desired. CAS [40] relies on user ability to remember images from their portfolio. During actual login, the user has to compute a path on a panel of images from top-left corner to the bottom-edge corner or right side of the panel based on whether the image on the panel at any point belongs to the user portfolio. The row or column at the bottom or right side of the panel has labels. When the user finishes the path, they have to input the label in response. The CAS scheme [40] is susceptible to SAT solver based attacks [19]. CAS also requires all $n = 80$ images to be shown at once similar to the APW scheme [5], which is impractical on small screens. The cognitive load of the scheme from Li and Teng [28] is very high as it requires the user to remember three different secrets and perform lexical-first matching on the challenge to obtain hidden sub-sequences. HB protocol [20] can be modified to use window based challenges, but it requires the user to add random responses with a skewed probability $\eta < \frac{1}{2}$, which can be hard for users. Foxtail protocol [26] reduces the response space to $\{0, 1\}$ at the expense of a high number of rounds for secure authentication. PAS [6] only resist a very small number of authentication sessions (< 10) [25]. The CHC scheme asks the user to locate at

least three pass-images in the challenge and click randomly within the imaginary convex hull of the pass-images. With the default parameter sizes $k = 5$ and $l = 82$ (on average), CHC is vulnerable to statistical attacks [42, 3] and usability is impacted with larger parameter sizes.Blum and Vempala [10] propose several simple cognitive schemes which are easy to compute for humans and require little training. Although their schemes are information theoretically secure, the guarantee is only for a small number of observed sessions (6 to 10). The scheme from Blocki et al [9] is provably secure against statistical adversaries, and can resist a sizeable number of observed sessions. The scheme's main drawback is the extensive training which requires a human user to memorize random mappings from 30-100 images to digits, which, even with memory aids such as mnemonics, could take considerable time. An interesting open question is to see if their proof strategy can be extended to show if BehavioCog is secure against statistical adversaries.

Various touch-based behavioural biometric schemes have been proposed for user authentication [41, 18, 24], which rely on simple gestures such as swipes. Simple gestures require a large number of samples to be collected to get good accuracy and are prone to observation attacks [23]. Sherman et al. [34] designed more complex (free-form) gestures, but which are only shown to resist human shoulder-surfing attacks. The closest work similar to ours is by Toan et. al. [32]. Their scheme authenticates users on the basis of how they write their PINs on the smartphone touch screen using x, y coordinates. In comparison, we do a more detailed feature selection process to identify features, which are repeatable and resilient against observation attacks. Furthermore, they report an equal error rate (EER) of 6.7% and 9.9% against random and shoulder-surfing attacks, respectively. Since these are EER values, the TPR is much lower than 1.0. To obtain a TPR close to 1.0, the FPR will need to be considerably increased. Thus, after observing one session, the observer has a non-negligible chance of getting in (since the PIN is no longer a secret). To achieve a low probability of random guess, the number of rounds in their scheme would need to be higher. Furthermore, after obtaining the PIN, the attacker may adaptively learn target user's writing by querying the authentication service. The use of a cognitive scheme, as mentioned before, removes this drawback. KinWrite [36], which asks the user to write their passwords in 3D space, and then authenticates them based on their writing patterns suffers from the same drawbacks. Pure graphical password schemes such as DéJà Vu [16], where the user has to click directly on pass-images or reproduce the same drawing on the screen, have the same vulnerability.

8 Discussion and Limitations

To begin, we show that a carefully designed training inspired by cognitive psychology helped users recognize their pass-emojis better. The potential of this needs to be further explored to see how large a set of images could be successfully recognized by users after longer gaps. A smaller number of pass-emojis is

also possible in our scheme at the expense of withstanding less observations; it may still be impractical for an attacker to follow a mobile user to record enough observations over a sustained period. Our results also show that users make themes to pick their pass-emojis. The top 15 pass-emojis chosen by all the users is shown in Table 10 (see E.5). The top 10 pass-emojis include 8 animals making animal the highest chosen category among the users. The other popular categories are food, fruits, transport and sports. Picking similar theme based images is a known challenge for graphical passwords and we left further exploration of issues arising due to the aforementioned challenge as a future research.

Behavioural biometrics tend to evolve over time and hence we see a slight increase in biometric errors after a week. A remedy is to frequently update the biometric template by replacing older samples [13]. On the flip side, we prefer behaviour biometrics over physiological biometrics due to this exact reason, since if stolen the consequences are less dire (user behaviour might evolve, words could be replaced, etc.). Additionally, the exact difficulty in mimicking cursively written words derived from certain English letters needs to be further explored (either experimentally or in theory). Also, the security of our scheme needs to be tested against a professional handwriting forger or a robot that can be programmed to mimic user gestures given video recordings, although we consider the latter to be a less likely attack in practice.

Our cognitive scheme might be susceptible to timing attacks [39] (c.f. Table 5). One way to circumvent this is to not allow the user to proceed unless a fixed amount of time has elapsed based on the highest average-time taken Finally, to protect the user's secret (pass-emojis and biometric templates), the authentication service could keep it encrypted and decrypt it only during authentication. A better solution requires the use of techniques such as fuzzy vaults [21] and functional encryption [12], and is left as future work.

9 Conclusion

The promise offered by cognitive authentication schemes that they are resistant to observation has failed to crystallize in the form of a workable protocol. Indeed, many researchers speculate that such schemes may never be practical. We do not refute this, but instead argue that combining cognitive schemes with other behavioural biometric based authentication schemes may make the hybrid scheme practical and still resistant to observation. Our scheme is not the only possibility. In fact, we need not confine ourselves to touch based biometrics, and may explore other behavioural biometric modalities. This way, several different constructions are conceivable.

References

[1] Asghar, H.J., Steinfeld, R., Li, S., Kaafar, M.A., Pieprzyk, J.: On the Linearization of Human Identification Protocols: Attacks Based on Linear Algebra, Coding Theory, and Lattices. IEEE TIFS 10(8), 1643–1655 (2015)

[2] Asghar, H.J., Kaafar, M.A.: When are Identification Protocols with Sparse Challenges Safe? The Case of the Coskun and Herley Attack. IACR's Cryptology ePrint Archive: Report 2015/1231 (2015)
[3] Asghar, H.J., Li, S., Pieprzyk, J., Wang, H.: Cryptanalysis of the Convex Hull Click Human Identification Protocol. Int. J. Inf. Secur. 12(2), 83–96 (2013)
[4] Asghar, H.J., Li, S., Steinfeld, R., Pieprzyk, J.: Does Counting Still Count? Revisiting the Security of Counting based User Authentication Protocols against Statistical Attacks. In: NDSS (2013)
[5] Asghar, H.J., Pieprzyk, J., Wang, H.: A New Human Identification Protocol and Coppersmith's Baby-step Giant-step Algorithm. In: ACNS. pp. 349–366 (2010)
[6] Bai, X., Gu, W., Chellappan, S., Wang, X., Xuan, D., Ma, B.: Pas: Predicate-based authentication services against powerful passive adversaries. In: ACSAC 2008. pp. 433–442 (2008)
[7] Baignères, T., Junod, P., Vaudenay, S.: How Far Can we Go Beyond Linear Cryptanalysis? In: Asiacrypt. pp. 432–450 (2004)
[8] Ballard, L., Lopresti, D., Monrose, F.: Forgery quality and its implications for behavioral biometric security. IEEE Transactions on Systems, Man, and Cybernetics 37(5), 1107–1118 (2007)
[9] Blocki, J., Blum, M., Datta, A., Vempala, S.: Towards Human Computable Passwords. In: ITCS (2017)
[10] Blum, M., Vempala, S.S.: Publishable humanly usable secure password creation schemas. In: Third AAAI Conf. on Human Computation and Crowdsourcing (2015)
[11] Bo, C., Zhang, L., Li, X.Y., Huang, Q., Wang, Y.: Silentsense: silent user identification via touch and movement behavioral biometrics. In: Mobicom. pp. 187–190 (2013)
[12] Boneh, D., Sahai, A., Waters, B.: Functional Encryption: Definitions and Challenges. In: TCC. pp. 253–273 (2011)
[13] Chauhan, J., Asghar, H.J., Kaafar, M.A., Mahanti, A.: Gesture-based Continuous Authentication for Wearable Devices: The Smart Glasses Use Case. In: ACNS. pp. 648–665 (2016)
[14] Coskun, B., Herley, C.: Can "something you know" be saved? In: ISC. pp. 421–440 (2008)
[15] Devijver, P.A., Kittler, J.: Pattern recognition: A statistical approach. PH (1982)
[16] Dhamija, R., Perrig, A.: DéJà Vu: A User Study Using Images for Authentication. In: Usenix Security. pp. 45–58 (2000)
[17] Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and Mining of Time Series Data: Experimental Comparison of Representations and Distance Measures. Proc. VLDB Endow. 1(2), 1542–1552 (2008)
[18] Frank, M., Biedert, R., Ma, E., Martinovic, I., Song, D.: Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. IEEE TIFS 8(1), 136–148 (2013)
[19] Golle, P., Wagner, D.: Cryptanalysis of a cognitive authentication scheme (extended abstract). In: SP. pp. 66–70 (2007)
[20] Hopper, N.J., Blum, M.: Secure Human Identification Protocols. In: Asiacrypt. pp. 52–66 (2001)
[21] Juels, A., Sudan, M.: A Fuzzy Vault Scheme. Des. Codes Cryptography 38(2), 237–257 (2006)
[22] Kao, H.S., Shek, D.T., Lee, E.S.: Control modes and task complexity in tracing and handwriting performance. Acta psychologica 54(1), 69–77 (1983)

[23] Khan, H., Hengartner, U., Vogel, D.: Targeted Mimicry Attacks on Touch Input Based Implicit Authentication Schemes. In: MobiSys '16. pp. 387–398 (2016)
[24] Li, L., Zhao, X., Xue, G.: Unobservable Re-authentication for Smartphones. In: NDSS (2013)
[25] Li, S., Asghar, H.J., Pieprzyk, J., Sadeghi, A.R., Schmitz, R., Wang, H.: On the Security of PAS (Predicate-Based Authentication Service). In: ACSAC. pp. 209–218 (2009)
[26] Li, S., Shum, H.Y.: Secure Human-Computer Identification (Interface) Systems against Peeping Attacks: SecHCI. Cryptology ePrint Archive, Report 2005/268
[27] Li, S., Ashok, A., Zhang, Y., Xu, C., Lindqvist, J., Gruteser, M.: Whose move is it anyway? Authenticating smart wearable devices using unique head movement patterns. In: PerCom. pp. 1–9 (2016)
[28] Li, X.Y., Teng, S.H.: Practical human-machine identification over insecure channels. Journal of Combinatorial Optimization 3(4), 347–361 (1999)
[29] Linial, N., Weitz, D.: Random vectors of bounded weight and their linear dependencies. http://dimacs.rutgers.edu/~dror/pubs/rand_mat.pdf (2000)
[30] Mandler, J.M., Johnson, N.S.: Some of the thousand words a picture is worth. Jrnl. of Exp. Psychology: Human Learning and Memory 2(5), 529–540 (1976)
[31] Matsumoto, T., Imai, H.: Human identification through insecure channel. In: EUROCRYPT. pp. 409–421 (1991)
[32] Nguyen, T.V., Sae-Bae, N., Memon, N.: Finger-Drawn PIN Authentication on Touch Devices. In: ICIP. pp. 5002–5006 (2014)
[33] Sakoe, H., Chiba, S.: A dynamic programming approach to continuous speech recognition. In: Seventh international congress on acoustics. vol. 3, pp. 65–69 (1971)
[34] Sherman, M., Clark, G., Yang, Y., Sugrim, S., Modig, A., Lindqvist, J., Oulasvirta, A., Roos, T.: User-generated Free-form Gestures for Authentication: Security and Memorability. In: MobiSys. pp. 176–189 (2014)
[35] Shokoohi-Yekta, M., Hu, B., Jin, H., Wang, J., Keogh, E.: Generalizing DTW to the multi-dimensional case requires an adaptive approach. Data Mining and Knowledge Discovery pp. 1–31 (2016)
[36] Tian, J., Qu, C., Xu, W., Wang, S.: Kinwrite: Handwriting-based authentication using kinect. In: NDSS (2013)
[37] Tversky, B., Sherman, T.: Picture memory improves with longer on time and off time. Jrnl. of Exp. Psychology: Human Learning and Memory 1(2), 114–118 (1975)
[38] Twitter, I., et al.: https://github.com/twitter/twemoji
[39] Čagalj, M., Perković, T.: Timing Attacks on Cognitive Authentication Schemes. IEEE TIFS 10(3), 584–596 (2014)
[40] Weinshall, D.: Cognitive Authentication Schemes Safe Against Spyware (Short Paper). In: SP. pp. 295–300 (2006)
[41] Xu, H., Zhou, Y., Lyu, M.R.: Towards Continuous and Passive Authentication via Touch Biometrics: An Experimental Study on Smartphones. In: SOUPS. pp. 187–198 (2014)
[42] Yan, Q., Han, J., Li, Y., Deng, R.H.: On Limitations of Designing Leakage-Resilient Password Systems: Attacks, Principles and Usability. In: NDSS (2012)

A Detailed Security Analysis of the Cognitive Scheme

Random Guess Attack: Let p_{RG} denote the success probability of a random guess. This probability is conditioned on the event $a \cap x$ being *empty* or not. Since this event

shall be frequently referred to in the text, we give it a special name: the *empty case*. Now the probability that i pass-objects are present in a, is given by $\mathbb{P}\left[|a \cap x| = i\right] = \binom{k}{i}\binom{n-k}{l-i}/\binom{n}{l}$, from which it follows that $\mathbb{P}\left[|a \cap x| = 0\right] \doteq p_0 = \binom{n-k}{l}/\binom{n}{l}$. We shall use the notation $\doteq$ when defining a variable. Thus, $p_{\text{RG}} = p_0 + (1-p_0)\frac{1}{d}$.

Brute Force Attack and Information Theoretic Bound: This attack is only possible after $\mathcal{A}$ has observed $m > 0$ challenge-response pairs (or samples) corresponding to successful authentication sessions. Before observing any samples, i.e., $m = 0$, all possible $\binom{n}{k}$ subsets are possible candidates of the target secret x. We denote a candidate by y, where quite possibly $y = x$. After observing one sample, the probability that a $y \in X$ is still a candidate for the secret x is given by $p_0 + (1-p_0)\frac{1}{d}$. Thus, we expect $\left(p_0 + (1-p_0)\frac{1}{d}\right)^m \binom{n}{k}$ subsets in X to still remain as candidates for x after observing m challenge-response pairs. Equating the above to 1, gives us $m \doteq m_{\text{it}} = -\log_2 \binom{n}{k} / \log_2(p_0 + (1-p_0)\frac{1}{d})$. We call m_{it}, the information theoretic bound on m. This is the least (expected) number of samples needed to be observed to obtain a unique candidate for the secret.

Meet-in-the-Middle Attack: This attack [20] works by first computing $\frac{k}{2}$-sized subsets of X on each of the m observed challenge-response pairs, and storing the m-element response string together with the subset in a hash table. After that, for each possible "intermediate" response string in $\mathbb{Z}_d^m$, and for each $\frac{k}{2}$-sized subsets of X we compute the final response string of m-elements. If this response string matches at least $m(1-p_0)$ responses[4] in the response string of the target secret x, we insert the intermediate response string together with the corresponding $\frac{k}{2}$-sized subset in the same hash table. Any collision in the hash table marks a possible candidate for x (by combining the two $\frac{k}{2}$-sized subsets). The time and space complexity of this attack is $\binom{n}{k/2}$.

Frequency Analysis: Frequency analysis is an attack proposed by Yan et al. [42][5] which could be done either independently or dependent on the response. In response-independent frequency analysis (RIFA), a frequency table of δ-tuples of objects is created, where $1 \leq \delta \leq k$. If a δ-tuple is present in a challenge, its frequency is incremented by 1. After gathering enough challenge-response pairs, the tuples with the highest or lowest frequencies may contain the k secret objects if the challenges are constructed with a skewed distribution. In the response-dependent frequency analysis (RDFA), the frequency table contains frequencies for each possible response in $\mathbb{Z}_d$, and the frequency of a δ-tuple is incremented by 1 in the column corresponding to the response (if present in the challenge).

First, note that our cognitive scheme is resistant to RIFA since the challenges are drawn uniformly at random without considering pass or decoy objects. This follows from Lemma 17 in [4]. To see that RDFA is also not applicable, define the indicator random variable $I(x')$ which is 1 if $x' \in a$, where $x' \subseteq x \in X$. We define a similar indicator random variable $I(y')$ for $y' \subseteq y \in X^{n-k}$, where X^{n-k} denotes the set of $n-k$ decoy objects. Now for RDFA to be inapplicable we should have $\mathbb{P}\left[I(x') = b \mid r = i\right] = \mathbb{P}\left[I(y') = b \mid r = i\right]$, for $i \in \mathbb{Z}_d$, $b \in \{0,1\}$ and $|x'| = |y'|$. Using Baye's rule $\mathbb{P}\left[I(x') = b \mid r = i\right] = \mathbb{P}\left[r = i \mid I(x') = b\right]\mathbb{P}\left[I(x') = b\right]/\mathbb{P}\left[r = i\right]$. Now, $\mathbb{P}\left[r = i\right] = p_0 \cdot \frac{1}{d} + (1-p_0)\frac{1}{d} = \frac{1}{d}$. Also, from Lemma 17 in [4] $\mathbb{P}\left[I(x') = 1\right] =$

[4] i.e., the expected number of samples that do not belong to the empty case.
[5] We borrow the term frequency analysis from [4].

$\mathbb{P}[I(y') = 1] = \binom{n-\delta}{l-\delta}/\binom{n}{l}$, where $\delta \doteq |x'| = |y'|$. From this, it follows that $\mathbb{P}[I(x') = 0] = \mathbb{P}[I(y') = 0]$. Now, $\mathbb{P}[r = i \mid I(y') = b] = \mathbb{P}[r = i] = \frac{1}{d}$, since the responses are not dependent on the decoy objects. Finally, we see that $\mathbb{P}[r = i \mid I(x') = 1] = \frac{1}{d}$, since at least δ pass-objects are present in the challenge, and the response is the sum modulo d, which due to the randomness of weights is distributed uniformly in $\mathbb{Z}_d$. If $I(x') = 0$, there are two possibilities. Either $\delta - 1$ or less number of pass-objects are present in the challenge, in which case the response is again uniform in $\mathbb{Z}_d$, or none of the pass-objects are present (empty case). But in the latter case, we ask the user to output a random response in $\mathbb{Z}_d$. Therefore, the probability of observing a response $r = i$ is $\frac{1}{d}$. From this it follows that our scheme is secure against RDFA.

Coskun and Herley Attack: Since only l objects are present in each challenge, the number of pass-objects present is also less than k with high probability. Let u denote the average number of bits of x used in responding to a challenge. The Coskun and Herley (CH) attack [14] states that if u is small, then candidates $y \in X, y \neq x$, that are close to x in terms of some distance metric, will output similar responses to x. If we sample a large enough subset from X, then with high probability there is a candidate for x that is a distance ξ from x. We can remove all those candidates whose responses are far away from the observed responses, and then iteratively move closer to x. The running time of the CH attack is at least $|X|/\binom{\log_2 |X|}{\xi}$ [14] where $|X| = \binom{n}{k}$, with the trade off that $m \approx \frac{1}{\epsilon^2}$ samples are needed for the attack to output x with high probability [2, 7]. The parameter ϵ is the difference in probabilities that distance $\xi + 1$ and $\xi - 1$ candidates have the same response as x. As we choose higher values of ξ, the complexity of the attack decreases but the probability differences become less prominent, which in turn means that more samples m need to be observed. The optimal value of ξ is when the time complexity is below a threshold, giving us a value of ϵ from which the number of required samples m can be obtained [2].

Linearization: We begin by assigning an order to the n objects in the global pool. We can then represent the secret x as an n-element binary vector $\mathbf{x}$ of Hamming weight k (where $x_i = 1$ indicates that object i is present in the secret). Similarly, a challenge $c = (a, w)$ can be represented by the n-element binary vector $\mathbf{a}$ of Hamming weight l (indicating the presence of the corresponding object) and the n-element vector $\mathbf{w}$ of Hamming weight $\leq l$, where $w_i = 0$ if $a_i = 0$. Let $\eta \in_U \mathbb{Z}_d$. Then our cognitive function f can be rewritten as $f(\mathbf{x}, \mathbf{c}) = b\mathbf{w} \cdot \mathbf{x} + \eta(1 - b) \bmod d$, where $b = \text{sgn}(\mathbf{a} \cdot \mathbf{x})$ is the sign function. Now, consider the case $r \doteq f(\mathbf{x}, \mathbf{c}) = 0$. This is possible if $b = 1$ and $\mathbf{w} \cdot \mathbf{x} \equiv 0 \bmod d$, or when $b = 0$ and $\eta = 0$. In the latter case, note that $\mathbf{w} \cdot \mathbf{x} = 0$ (even without the modulus), and hence trivially $\mathbf{w} \cdot \mathbf{x} \equiv 0 \bmod d$. On the other hand, if $r \neq 0$, we again have the possibility that if $b = 1$, $\mathbf{w} \cdot \mathbf{x} \equiv r \bmod d$ or if $b = 0$, then $\eta = r$. However, we cannot write the latter as an equation in $\mathbf{x}$ and $\mathbf{w}$ without including the non-zero noise term η.

Thus one attack strategy is to keep samples corresponding to a 0 response to build a system of linear congruences. After n such congruences have been obtained, $\mathcal{A}$ can use Gaussian elimination to obtain a unique solution for $\mathbf{x}$, thus obtaining the secret. That is, create the matrix W whose ith row corresponds to the weight vector from the ith challenge $\mathbf{c}_i$ such that the corresponding response is 0. This gives us the system of linear congruences $W\mathbf{x} \equiv \mathbf{0} \bmod d$, where W is an $n \times n$ square matrix. Of course, W needs to be a full rank matrix. This can be done by observing a little over n samples (with 0 response), because with high probability a randomly generated W is of full

rank if l is large enough [2, 29]. For instance, with $(k, l, n) = (14, 30, 140)$ we found that a fraction 0.29 of the matrices generated had full rank by running a Monte Carlo simulation with 10,000 repetitions. Note that since the response is uniformly distributed in $\mathbb{Z}_d$, we expect to construct W after observing dn challenge-response pairs. Thus, we are discarding all challenges that correspond to a non-zero response.

Another way of linearization that does not discard any challenges, but requires the observations of the same number of challenge-response pairs, is to introduce $(d-1)n$ new binary variables. We illustrate this using $d = 2$ as an example. Let $\mathbf{w}_i$ denote the ith n-element weight vector. Then we can form the system

$$\begin{pmatrix} \mathbf{w}_1 & 1\,0\cdots 0 \\ \mathbf{w}_2 & 0\,1\cdots 0 \\ \vdots & \vdots\,\vdots\,\ddots\,\vdots \\ \mathbf{w}_n & 0\,0\cdots 1 \\ \mathbf{w}_{n+1} & 0\,0\cdots 0 \\ \vdots & \vdots\,\vdots\,\ddots\,\vdots \\ \mathbf{w}_{2n} & 0\,0\cdots 0 \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \\ x_{n+1} \\ \vdots \\ x_{2n} \end{pmatrix} \equiv \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \bmod 2,$$

where $x_{n+1}, \ldots, x_{2n}$ are n new variables. The above system of equations is obtained by observing $2n$ challenge-response pairs and re-arranging the 0 and 1 responses (the top n rows correspond to $r = 1$). Let us call the $2n \times 2n$ matrix, W. By construction of the last n columns of W, the $2n$ rows of W are linearly independent regardless of the vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n$ as long as the vectors $\mathbf{w}_{n+1}, \ldots, \mathbf{w}_n$ remain linearly independent. But we have seen above that this is true with high probability. Hence, we can use Gaussian elimination again to uniquely obtain the secret. To see that the above system is consistent with observation, consider the first row. If it corresponds to the empty case, then by setting $x_{n+1} = 1$ we get the response 1. On the other hand, if it is not the zero case then $x_{n+1} = 0$ satisfies the equation. Any of the two values of x_{n+1} satisfy the 0-response rows. Since the responses are generated randomly, we expect to obtain the above system by observing dn challenge-response pairs. Note that if $\mathcal{U}$ were to respond with 0 in the empty case, then we could obtain a linear system of equations after n challenge-response pairs. The introduction of noise expands the number of required challenge-response pairs to dn, an increase by a factor of d. Gaussian elimination is by far the most efficient attack on our scheme, and therefore this constitutes a significant gain.

Generalization: With the exception of Gaussian elimination, all other attacks mentioned above have complexity exponential in one or more variables in (k, l, n). Since the above linearization works after observing dn challenge-response pairs, we believe the problem of finding a polynomial time algorithm in (k, l, n) which uses $m < dn$ number of samples (say $(d-1)n$ samples) from the function described in Eq. 1 is an interesting open question.

B Feature Comparison

Figure 3 shows the feature `x`, i.e., the x-coordinate, as a time series for the complex word "`xman`." For ease of view, we show the feature without normalization. Observe that the way the word is written varies between different users (two samples from User 1, User 2 and User 3), while remains similar for the same user (User 1-A and 1-B).

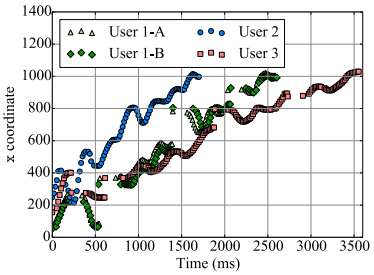


Fig. 3: Feature comparison of samples from three users.

C Feature Selection Algorithms

Algorithm 1: Select Features

input: Set of all features Q_{tot}, a symbol $\in \Omega$, a set of user-attacker pairs $(\mathcal{U}, \mathcal{A})$.

Initialize $Q_{\text{sel}}^{(0)} \leftarrow \emptyset$, $i \leftarrow 0$.
for $j = 1$ *to* $|Q_{\text{tot}}|$ **do**
    Set $i \leftarrow i + 1$.
    Create temporary feature subsets Q_j by adding feature $q_j \in Q_{\text{tot}}$ to $Q_{\text{sel}}^{(i-1)}$.
    for *each* $(\mathcal{U}, \mathcal{A})$ *pair* **do**
        Run Get z-List algorithm (Algorithm 2) with inputs Q_j, $\mathcal{U}$ and $\mathcal{A}$ to get a z-list.
    Sum TPR and FPR values for all users for each value of $0 \leq z \leq z_{\max}$ in the z-list.
Let Q_j be the temporary feature subset that has the minimum FPR sum with TPR sum equal to 1.0.
Set $Q_{\text{sel}}^{(i)} \leftarrow Q_{\text{sel}}^{(i-1)} \cup \{q_j\}$, $Q_{\text{tot}} \leftarrow Q_{\text{tot}} - \{q_j\}$.
Repeat Steps 2-9 until Q_{tot} is empty.
Return $Q \doteq Q_{\text{sel}}$ from the $Q_{\text{sel}}^{(i)}$'s that has the least FPR.

Algorithm 2: Get z-List

input: Feature subset Q, registration and test samples from $\mathcal{U}$, test samples from $\mathcal{A}$.

For the features in Q, find the optimal template $\hat{Q}$ together with μ and σ from $\mathcal{U}$'s registration samples.
Initialize an empty z-list.
Initialize $z \leftarrow 0$, step $\leftarrow 0.125$, TP $\leftarrow 0$, FP $\leftarrow 0$.
while $z \leq z_{\max} \doteq 10$ **do**
    Set $\hbar \leftarrow \mu + z\sigma$
    for *each test sample from* $\mathcal{U}$ **do**
        if DTW *distance between* $\hat{Q}$ *and test sample is* $\leq \hbar$ **then**
            TP $\leftarrow$ TP $+ 1$.
    for *each test sample from* $\mathcal{A}$ **do**
        if DTW *distance between* $\hat{Q}$ *and test sample is* $\leq \hbar$ **then**
            FP $\leftarrow$ FP $+ 1$.
    Compute TPR and FPR by normalizing the TP and FP values.
    Update z-list with the tuple $(z, \text{TPR}, \text{FPR})$.
    Set $z \leftarrow z +$ step.
Return z-list.

D Training Game Design

The training game consists of the following steps:

1. The user was shown a fixed number (initially 5) of emojis without weights. At least one of them was a (random) pass-emoji. The user was told exactly how many of their pass-emojis were present and was asked to tap on them. Gradually, the emojis were increased from 5 to 25 in steps of 5 (with a corresponding increase in pass-emojis).
2. To help the user associate responses in $\mathbb{Z}_5$ to words, mnmonic associations were shown to the user as shown in Table 6. The mnemonic strategy used is a mixture of the (rhyming) peg method, keyword method and picture-based mnemonics [9, 10]. The user was then given a series of easy questions with the correct answer being the complex word. An example question was: "0 rhymes with hero, who is our hero? `xman` or batman?" There were three different questions for each word, meaning the user wrote each word three times.
3. This step was the same as Step 1 except that (a) the user was not told how many of their pass-emojis were present, and (b) the number of images was increased from 5 to 30 in steps of 5.
4. This step was the same as Step 3 except that (a) the images also had weights in $\mathbb{Z}_5$, and (b) the user had to compute f, map the response to the word and press one of five buttons corresponding to the correct word.
5. This step was the same as Step 2 except that the questions asked were slightly more difficult, e.g., "0 rhymes with hero, _ _ _ _ is our hero." The user had to write each symbol two more times.

response	mnemonic	word
0	hero	`xman` is our hero
1	run	`bmwz` runs on the street
2		the duck goes `quak`
3		I got `hurt` by a trident
4		can't see four when it's `fogy`

Table 6: Mnemonic mapping of cognitive response to complex words.

E Extended Results

E.1 Symbol Set Analysis

To dig deeper into why some symbol sets have poorer average FPR than others, we did some further analysis. For the worst case scenario, we did the following for each symbol: First, we fix $z = 1$, pick best features for each symbol through the feature selection algorithm and pick first 10 biometric training samples from Session 1 to train the classifier for each user. Next, we tested the classifier, (a) using user's own last three samples from Session 1 to obtain TPR values, (b) using user's three samples from Session 2 to obtain TPR values, and (c) using attacker's three samples from Session 2 to obtain FPR values. The results are shown in Table 7. The average TPR for all users for Session 1 is denoted TPR_1, whereas for Session 2 is denoted TPR_2. We can see that the average TPR for Session 2 decreases from Session 1 for complex figures drastically which means that users find it hard to repeat drawings of complex figures. A near consistent average TPR but a high average FPR between the two sessions for easy words and easy figures means they are repeatable but not secure against video based observation attacks. The reason for easy words to be easily mimicked is because of the presence of letters, which do not contain many sharp turns such as *o*, *c* and *s*. The easy figures were easily attacked because drawing them does come naturally to the users and hence they draw them slowly, which makes it easy for an attacker to pick and then mimic. The results for complex words show that they are both highly repeatable and cannot be easily mimicked. Users can write words fluently (due to years of practice), thereby making them difficult to be mimicked.

E.2 Training Time

Table 8 shows the time taken by different user groups in completing the training. Users in both Group 2 and Group 3 spend 50% of their total training time for the cognitive scheme to familiarize themselves with their pass-emojis and also learning how to use the scheme. The time to collect biometric samples takes 50%, 30%, and 37% of the total training time for Group 1, 2 and 3, respectively.

Table 7: Results indicating repeatability and resilience against observation attacks for different symbol sets.

Symbol Category	TPR_1 (average)	TPR_2 (average)	Average FPR
easy words	0.93	1.00	0.24
complex words	0.91	1.00	0.05
easy figures	0.68	0.60	0.21
complex figures	0.70	0.53	0.39

Table 8: Average registration time (seconds) of different user groups in Phase 2.

Group	Pass-emojis selection time	Cognitive training time	Biometric training time	Total training time
1	128	0	129	257
2	114	284	174	573
3	105	359	282	746

E.3 Distribution of Symbols in the Empty Case

In the empty case, the user is supposed to write a random complex word. We want to see if the resultant distribution of symbols is random or not. We had a total of $v = 34$ instances of the empty case. The probability of randomly choosing a word is $\frac{1}{d} = \frac{1}{5}$. The number of times, i, a word was written in a total of v empty cases, is once again binomially distributed (conditioned on the null hypothesis) with this probability. We consider $i \leq 3$ and $i \geq 11$ as statistically significant (as they imply $p < 0.05$). The word `xman` (corresponding to $r = 0$) was significantly overused ($i = 13$), whereas the word `bmwz` (mapped to $r = 1$) was significantly less used ($i = 2$). The frequency of occurrence of other words did not deviate (statistically) significantly. We believe the reason for overuse of `xman` might be because the user thought that an empty case implies the cognitive response is 0. Note that the users were told that they need to write *any* word in the empty case.

E.4 Effect of Number of Pass-Emojis Present

The percentage of cognitive errors and authentication time increases with an increasing number of pass-emojis present in the challenge (Table 9). The time taken in the empty case is more than the time taken when one or more pass-emojis is present. This might be because the user needs more time to ensure if it is indeed the empty case.

E.5 Pass-Emojis Chosen by Users

The probability that an emoji is present in a random sample of k emojis out of n is $\frac{k}{n}$. Thus in v random samples, the probability that an emoji occurs i times is given by $p \doteq \binom{v}{i} \left(\frac{k}{n}\right)^i \left(1 - \frac{k}{n}\right)^{v-i}$. Setting $v = 30$ (30 users) in the above, we see that if an emoji occurs $i \geq 6$ times in the 30 chosen pass-emojis, we consider the event statistically significant ($p < 0.05$).[6] Our results show that 15 emojis were selected by at least six

[6] For the lower tail, we see that the probability is always higher than 0.05 since v is small.

Table 9: Time taken and percentage of cognitive errors a function of number of pass-emojis present in a challenge.

# of pass-emojis	Frequency	Average time (sec)	Cognitive errors (%)
0	34	18.08	0.00
1	83	14.63	36.14
2	111	17.52	42.34
3	67	16.90	44.77
4	40	18.30	50.00
5	21	22.60	52.38
6	4	16.22	25.00

or more users. Ten of the 15 emojis are animals, which seems to indicate that users were choosing their pass-emojis using an animal theme. This is perhaps also due to the fact that animals constituted a high percentage of the total emojis. The 15 emojis are shown in Table 10 along with the number of users who chose them.

Frequency	Emojis
9	
8	
7	
6	

Table 10: The 15 most popular emojis in users' pass-emojis.

E.6 Recognizing Pass-Emojis

The minimum, maximum and average number of pass-emojis recognized was, respectively, (7, 12, 9.0) for Group 1, (8, 13, 10.5) for Group 2 and (10, 14, 12.1) for Group 3. These results were obtained by asking the users to select their pass-emojis from the total pool of emojis after a gap of one week. If the user does not remember any of the pass-emojis, the probability of correctly selecting i out of k emojis is given by $p \doteq \binom{k}{i}\binom{n-k}{k-i}/\binom{n}{k}$. An $i \geq 4$ is significant (since $p < 0.05$). We can see that all groups were able to remember a significant number of their pass-emojis. More training may help users in recognizing their pass-emojis in the longer term. However, the higher recognition rates for both Groups 2 and 3 do not translate into higher successful authentication attempts in Session 2. We investigate this issue in Phase 3. We also conclude that without much training, users may easily recognize around up to 7 emojis even after a gap in time.

E.7 Guessing Pass-Emojis

Here we consider that if an attacker can guess more than 4 pass-emojis of the target user, then the attacker has significant advantage over random guess. Five of the 30

attackers were able to guess 4 or more pass-emojis of the target user, and one attacker guessed as many as 11 as the attacker thought that the target user might have picked pass-emojis according to a animal based theme. Picking theme based pass-emojis might lead to more chances of being successfully attacked.

E.8 Questionnaire Statistics

At the end of Session 2, we asked the (30) users to fill a questionnaire on a Likert scale of 1 to 5, where 1 means Strongly disagree, 2 means Disagree, 3 means Neither Agree or Disagree, 4 means Agree and 5 means Strongly Agree. Overall, the results indicate that, (a) users find rendering the words to be easy on smartphone, (a) users liked playing the training game, and (c) users think that the number of pass-emojis (14) is high. For more details, please refer to Appendix.

The general consensus about the ease of writing words on the smartphone screen was a rating of 4. The users liked playing the training game during the registration with an overall rating of 4. The overall usability of the scheme received mixed rating from the users (3). However, the users who mostly rated 1 or 2 for usability said that they are likely to use the system if it can provide a high security guarantees. The major issue the users had with our scheme is the number of pass-emojis. The rating was 2 when the users were asked if they could manage 14 pass-emojis easily. A 53% of the users say that they would not like to use the system because of 14 pass-emojis, and 30% users found it hard to recognize their pass-emojis during authentication. Only 16% and 6% of the users complained about entering biometric responses and computing f.

Most users (53%) prefer 6-10 pass-emojis as their secret followed by 30% who prefer no more than 5 pass-emojis. In response to the question on the size of l (i.e., the window size), 53% of the users responded with 0-10 emojis. 23% users said 11-20 and a similar percentage were fine with the current scheme (21-30 emojis). When asked how they picked their pass-emojis, 18 users said they created a certain theme to make it easy for them to remember their pass-emojis. Some users used multiple themes; 7 users said that they picked animals, 6 users picked food items, 2 users picked tools, 2 users picked sports, one picked recreation and one picked faces. Two users created a theme based on a story. One story was: “Santa watching sports while eating a lot of food.”