

BehavioCog: An Observation Resistant Authentication Scheme

Jagmohan Chauhan^{1,2}(✉), Benjamin Zi Hao Zhao¹, Hassan Jameel Asghar², Jonathan Chan², and Mohamed Ali Kaafar²

¹ UNSW, Sydney, Australia

{jagmohan.chauhan, ben.zhao}@data61.csiro.au

² Data61, CSIRO, Sydney, Australia

{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 s 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 [30], 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 [5, 20, 26, 39]. 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) [39] requires 11 rounds resulting in 120 s to authenticate,

The full (more detailed) version is available as the conference version of this paper.

while the Hopper and Blum (HB) scheme [20] requires 20 rounds and 660s [41]. An authentication time between 10 to 30s 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. 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 $FPR \times |R|^{-1} = 0.10 \times 0.25 = 0.025$. 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 minimizes the authentication time and reduces the 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.

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 [31, 33] (free-hand drawings) only tackle shoulder-surfing attacks, and (b) the cognitive schemes proposed in the literature are either not secure [39] against known attacks or not usable due to high cognitive load (see Sect. 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 [1, 26]. 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 19s, 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

2.1 Preliminaries

Authentication Schemes: A *shared-secret challenge-response* authentication scheme consists of two protocols: *registration* and *authentication*, between the 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 $f : X \times C \rightarrow R$ as the cognitive function. It has to be computed mentally by the user. The 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 an indefinite access to the video recordings of the user touch gestures and tries to mimic the user. Unlike the original threat model from Matsumoto and Imai, our threat model assumes that the device as well as the communication channel between the device and \mathcal{S} are secure.

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 $\text{sym}(r)$. Upon receiving the rendering of $\text{sym}(r)$, \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 alphanumeric). 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 i th weight in w is denoted w_i and corresponds to the i th 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

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

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 $\text{sym} : \mathbb{Z}_d \rightarrow \Omega$ is the straightforward lexicographic mapping and is public. We assume a $(d + 1)$ -classifier D (see Sect. 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 Fig. 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 $\text{sym}(r)$, 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.	2: Authentication.
1 \mathcal{U} and \mathcal{S} share a secret $x \in X$.	1 \mathcal{S} sets $\text{err} = 0$.
2 For each symbol in Ω , \mathcal{U} sends t renderings to \mathcal{S} .	2 for γ rounds do
3 For each symbol in Ω , \mathcal{S} trains D on the t renderings to obtain \mathcal{U} 's template.	3 \mathcal{S} samples $c = (a, w) \in_U C$ and sends it to \mathcal{U} .
4 The secret consists of x and the d templates.	4 \mathcal{U} computes $r = f(x, c)$.
	5 \mathcal{U} renders the symbol $\text{sym}(r)$, and sends it to \mathcal{S} .
	6 \mathcal{S} runs D on the rendering.
	7 if D outputs “none” then
	8 \mathcal{S} sets $\text{err} = 1$.
	9 else
	10 \mathcal{S} obtains r corresponding to the symbol output by D .
	11 if $x \cap a \neq \emptyset$ and $r \neq f(x, c)$ then
	12 \mathcal{S} sets $\text{err} = 1$.
	13 if $\text{err} = 1$, \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) [1, 26]. 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. [41] outlined in Sect. 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. We next discuss the security of our cognitive scheme.

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 in our full paper. 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, $p_{\text{RG}} = p_0 + (1 - p_0)\frac{1}{d}$.

Brute Force Attack (BF) and Information Theoretic Bound. This attack outputs a unique candidate for the secret after $m \doteq m_{\text{it}} = -\log_2 \binom{n}{k} / \log_2(p_0 + (1 - p_0)\frac{1}{d})$ 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 $\binom{n}{k/2}$. 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. [41],³ could be done either independently or dependent on the response. In response-independent frequency analysis, 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, 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). Our scheme is immune to both forms of frequency analysis (see Appendix A of the full paper).

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, then Gaussian elimination can be used to uniquely obtain the secret. In Appendix A of our full paper, 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

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

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).

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.

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.

(d, k, l, n)	m_{it}	p_{RG}	BF	MitM	CH	GE	(d, k, l, n)	γ	Sessions (med. \mathcal{A})	Sessions (high \mathcal{A})	Ω	Security
$(5, 5, 24, 60)$	11	0.255	2^{22}	2^{12}	2^{11}	poly(n)	$(5, 5, 24, 60)$	1	10	10	CW	1.3×10^{-2}
Samples required	-	0	11	11	23	300	$(5, 5, 24, 60)$	2	5	5	CW	1.5×10^{-4}
$(5, 10, 30, 130)$	24	0.252	2^{48}	2^{28}	2^{33}	poly(n)	$(5, 5, 24, 60)$	3	3	3	CW	2×10^{-6}
Samples required	-	0	24	24	24	650	$(5, 10, 30, 130)$	1	24	24	CW	1.3×10^{-2}
$(5, 14, 30, 180)$	34	0.256	2^{68}	2^{40}	2^{40}	poly(n)	$(5, 10, 30, 130)$	2	12	12	CW	1.5×10^{-4}
Samples required	-	0	34	34	94	900	$(5, 10, 30, 130)$	3	8	8	CW	2×10^{-6}
$(5, 18, 30, 225)$	44	0.254	2^{87}	2^{51}	2^{51}	poly(n)	$(5, 14, 30, 180)$	1	94	34	CW	1.3×10^{-2}
Samples required	-	0	44	44	168	1125	$(5, 14, 30, 180)$	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}

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 used complex words in the implementation of our scheme as it was the best symbol set. The words or figures are used for behavioural biometrics while emojis are used for cognitive scheme.

Easy words: These are 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 [13, 24], contain sharp turns

and angles [33], the users finger must move in all directions while drawing the symbol. To help the user associate responses in \mathbb{Z}_5 to complex words, mnemonic associations were used (Appendix D in the full paper).

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

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

4.2 Choice of Classifier

We picked dynamic time warping (DTW) [32] 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, 31] (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 [32]. 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 [34]: $\text{DTW}(\hat{Q}, Q) = \sum_{i=1}^{|\hat{Q}|} \text{DTW}(\hat{q}_i, q_i)$, 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}_{sym} to check if \mathcal{U} produced a valid rendering of a symbol from Ω (only using x, y coordinates) and (2) \hat{Q}_{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 $\bar{h} \doteq \mu + z\sigma$. 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 \bar{h}_{sym} and \bar{h}_{user} correspond to \hat{Q}_{sym} and \hat{Q}_{user} , respectively. The classification works as follows:

Step 1: If for a given challenge $c = (a, w)$, $x \cap a \neq \emptyset$ (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}_{sym} and the sample is $> \bar{h}_{\text{sym}}$. Otherwise, \mathcal{S} moves to Step 2. In the empty case, \mathcal{S} computes the DTW distance between the sample and \hat{Q}_{sym} for each symbol and picks the symbol which gives the least distance. Next, the distance is compared with \bar{h}_{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}_{user} of the symbol. If the distance is $> \bar{h}_{\text{user}}$, the user is rejected, otherwise it is accepted.

4.5 Feature Identification and Selection

We identify 19 types of features from the literature [11, 13, 35, 40] and obtain 40 features (Table 3), most of which are self explanatory. Explanation of curvature, slope angle and path angle is described in [35]. Device-interaction features were obtained using the inertial motion sensors: accelerometer and gyroscope of the smartphone. 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 in the full paper illustrates the discriminatory power of a single feature (\mathbf{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 of our full paper. 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 $\text{TPR} = 1.0$ and minimizing the FPR values (calculated based on user-adversary pairs, see Sect. 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 $\text{TPR} = 1.0$ and the least FPR as the final set of features. The algorithm calls the Get z -List algorithm (Algorithm 2 in Appendix C of our full paper) 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	$x, y, \delta x, \delta y$	Top, bottom, left, right most point	TMP, BMP, LMP, RMP	Rotational position of device in space	R_x, R_y, R_z
Velocity along coordinates	\dot{x}, \dot{y}	Width: RMP – LMP, Height: TMP – BMP	width, height	Rate of rotation of device in space	G_x, G_y, G_z
Acceleration along coordinates	\ddot{x}, \ddot{y}	Rectangular area: width \times height	area	3D acceleration force due to device’s motion and gravity	A_x, A_y, A_z
Pressure and change in pressure	$p, \delta p$	Width to height ratio	WHR	3D acceleration force solely due to gravity	ξ_x, ξ_y, ξ_z
Size and change in size	$s, \delta s$	Slope angle	θ_{slope}	3D acceleration force solely due to device’s motion	a_x, a_y, a_z
Force: $p \times s$	F	Path angle	θ_{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* [37]. We used the parameters $(k, l, n) = (14, 30, 180)$ (corresponding to the medium strength adversary, see Sect. 3.2). FastDTW was used to implement DTW [32] with radius 20. More details are available in our full paper.

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 of our full paper). 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 10s followed by a 3s cool off period (inspired by cognitive psychology literature [29, 36]), 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

Phase 1. We find the best symbol set in terms of repeatability and security by selecting features (Algorithm 1, Appendix C of the full paper) 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. More analysis is in Appendix E.1 of our full paper.

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	$x, y, \delta x, \delta y, \text{TMP}, \theta_{\text{slope}}, \theta_{\text{path}}, R_x$	$\text{TMP}, \text{height}, \text{WHR}, \theta_{\text{slope}}, \theta_{\text{path}}$
Complex words	0.00	0.05	$y, \delta y, p, \text{height}, \text{area}, \theta_{\text{slope}}, R_y$	$\delta x, \text{height}, \theta_{\text{path}}$
Easy figures	0.01	0.38	$y, \delta x, \delta y, p, F, \text{height}, \text{area}, \theta_{\text{slope}}, \theta_{\text{path}}$	$y, \delta y, p, \text{height}$
Complex figures	0.01	0.39	δx	$x, \text{TMP}, \text{BMP}$

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 and attacks. More results are in Appendix E of our full paper.

Registration Time: The average time to select 14 pass-emojis was around 2 min for all groups. The maximum training time was 12 min for Group 3, since it had the most amount of training, and the minimum was 4 min 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 (Appendix E.8 of our full paper). Detailed results regarding registration are shown in Appendix E.2 of our full paper.

Authentication Time: Table 5 shows the average authentication time (per round) taken by different user groups in the two sessions. Generally, the user spends 15–20s in computing f and 6–8s 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	Av. Biometric	Av. Processing	Av. Total	Success	Cognitive	Biometric
	Time (sec)	Time (sec)	Time (sec)	Time (sec)	Rate (%)	Errors (%)	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. Phase 3 was done to find out the cause for a high number of cognitive errors.

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]).

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. During actual login in CAS [39], 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 [39] is susceptible to SAT solver based attacks [19]. CAS also uses parameter sizes of $n = 80$ and $k = 30$, and all n images need to be shown on the screen at once, which is hard to display on touch screens of smartphones. 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 [3, 41] and usability is impacted with larger parameter sizes. Blum et al. [10] propose simple cognitive schemes which are easily human computable but are information theoretically secure for only 6 to 10 observed sessions. The scheme from Blocki et al. [9] is provably secure against statistical adversaries and can resist a sizeable number of observed sessions. However, their scheme's require extensive training.

Various touch-based behavioural biometric schemes have been proposed for user authentication [18, 24, 40], 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. [33] 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. [31]. 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 removes this drawback. KinWrite [35], 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

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. We also show that users make themes to pick their pass-emojis (Appendix E.5 in our full paper). Issues arising due to picking similar theme based images is left 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 proposed scheme is to be tested against a professional handwriting forger or a sophisticated robot who can be programmed to mimic gestures. Our cognitive scheme might be susceptible to timing attacks [38] (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 can use techniques such as fuzzy vaults [21] and functional encryption [12], and is left as a 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. Many researchers speculate that such schemes may never be practical. We do not deny

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, in addition to touch based biometrics other behavioural biometric modalities can be explored. 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: Zhou, J., Yung, M. (eds.) *ACNS 2010*. LNCS, vol. 6123, pp. 349–366. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13708-2_21
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: Lee, P.J. (ed.) *ASIACRYPT 2004*. LNCS, vol. 3329, pp. 432–450. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30539-2_31
8. Ballard, L., Lopresti, D., Monrose, F.: Forgery quality and its implications for behavioral biometric security. *IEEE Trans. Syst. Man Cybern.* **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 Conference 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: Ishai, Y. (ed.) *TCC 2011*. LNCS, vol. 6597, pp. 253–273. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-19571-6_16
13. Chauhan, J., Asghar, H.J., Mahanti, A., Kaafar, M.A.: Gesture-based continuous authentication for wearable devices: the smart glasses use case. In: Manulis, M., Sadeghi, A.-R., Schneider, S. (eds.) *ACNS 2016*. LNCS, vol. 9696, pp. 648–665. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39555-5_35
14. Coskun, B., Herley, C.: Can “something you know” be saved? In: Wu, T.-C., Lei, C.-L., Rijmen, V., Lee, D.-T. (eds.) *ISC 2008*. LNCS, vol. 5222, pp. 421–440. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85886-7_29

15. Devijver, P.A., Kittler, J.: *Pattern Recognition: A Statistical Approach*. Prentice-Hall, Englewood Cliffs (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: Boyd, C. (ed.) *ASIACRYPT 2001*. LNCS, vol. 2248, pp. 52–66. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_4
21. Juels, A., Sudan, M.: A fuzzy vault scheme. *Des. Codes Crypt.* **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 Psychol.* **54**(1), 69–77 (1983)
23. Khan, H., Hengartner, U., Vogel, D.: Targeted mimicry attacks on touch input based implicit authentication schemes. In: *MobiSys 2016*, 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. *J. Comb. Optim.* **3**(4), 347–361 (1999)
29. Mandler, J.M., Johnson, N.S.: Some of the thousand words a picture is worth. *J. Exp. Psychol. Hum. Learn. Mem.* **2**(5), 529–540 (1976)
30. Matsumoto, T., Imai, H.: Human identification through insecure channel. In: Davies, D.W. (ed.) *EUROCRYPT 1991*. LNCS, vol. 547, pp. 409–421. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-46416-6_35
31. Nguyen, T.V., Sae-Bae, N., Memon, N.: Finger-drawn PIN authentication on touch devices. In: *ICIP*, pp. 5002–5006 (2014)
32. Sakoe, H., Chiba, S.: A dynamic programming approach to continuous speech recognition. In: *Seventh International Congress on Acoustics*, vol. 3, pp. 65–69 (1971)
33. 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)
34. Shokoohi-Yekta, M., Hu, B., Jin, H., Wang, J., Keogh, E.: Generalizing DTW to the multi-dimensional case requires an adaptive approach. *Data Min. Knowl. Discov.* **31**, 1–31 (2016)
35. Tian, J., Qu, C., Xu, W., Wang, S.: KinWrite: handwriting-based authentication using kinect. In: *NDSS* (2013)

36. Tversky, B., Sherman, T.: Picture memory improves with longer on time and off time. *J. Exp. Psychol. Hum. Learn. Mem.* **1**(2), 114–118 (1975)
37. Twitter, I., et al.: <https://github.com/twitter/twemoji>
38. Čagalj, M., Perković, T.: Timing attacks on cognitive authentication schemes. *IEEE TIFS* **10**(3), 584–596 (2014)
39. Weinshall, D.: Cognitive authentication schemes safe against spyware (Short Paper). In: SP, pp. 295–300 (2006)
40. 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)
41. Yan, Q., Han, J., Li, Y., Deng, R.H.: On limitations of designing leakage-resilient password systems: attacks, principles and usability. In: NDSS (2012)