POSTER: A Behavioural Authentication System for Mobile Users

Md Morshedul Islam and Reihaneh Safavi-Naini
University of Calgary, Canada

ABSTRACT

Active *behavioural-based authentication* systems are challenge-response based implicit authentication systems that authenticate users using the behavioural features of the users when responding to challenges that are sent from the server. They provide a flexible (no extra hardware) and secure second factor for authentication systems, with applications including protection against identity theft and password compromise of web applications. We propose a novel active behavioural authentication system for mobile devices, called **DAC** (Draw A Circle), where a challenge specifies a set of constraints on a circle and the response is a user drawn circle that satisfies the constraints. We carefully select a set of features that capture behavioural traits of the user which is used to construct a profile for them, then design a matching algorithm that allows users to be authenticated with approximately 95% accuracy. We discuss our implementation, and present our experimental results that show, (i) the accuracy of authentication system and (ii) non-delegateability of profile, guaranteeing that the user cannot pass their credentials to others.

Keywords

User authentication, Behavioural authentication, Challenge-response, Mobile authentication

1. INTRODUCTION

Mobile devices are an integral part of our everyday life. They are used for electronic commerce, accessing governmental and financial services, and participation in entertainment and social networking sites. The first step in all applications is user authentication, which today is primarily in the form of passwords. Two major drawbacks of password systems are vulnerability to (i) *password theft*, and (ii) *being passed on*. This latter is when a user shares their password with a third party to allow them to access their account, for example in accessing a company's network, or one's subscription services.

CCS'16 October 24-28, 2016, Vienna, Austria

ACM ISBN 978-1-4503-4139-4/16/10.

DOI: http://dx.doi.org/10.1145/2976749.2989065

To alleviate these problems *implicit authentication* [5, 3, 1, 2] have been used as a second authentication factor. In these systems, a model of user behaviour is constructed using data that is captured from various sensors and is used to authenticate users. Implicit authentication does not need extra hardware such as a secureID token, and can substantially increase the confidence about the authentication decision. It has two phases: a *registration phase* during which a profile of the user is constructed, and an *authentication phase* during which the user provides data to the system that will be matched against the stored profile. A profile consists of a set of *features* such as user locations, or websites visited by them, or their behavioural biometrics (keystroke dynamic and typing pattern). Those features could be highly predictable and possibly shareable. *Active behavioural systems* [1] are challenge and response systems where a user profile consists of features that are measured during a well-designed activity. During authentication phase, a challenge that is a request for an activity is sent to the user, and the response is matched against the stored profile.

We propose a novel active behavioural authentication system, called DAC (Draw A Circle), for mobile devices that utilize the device sensor to capture users' behavioural traits such as short-term memory, drawing speed and accuracy, and their physical characteristics to construct a set of robust features that are not easily learnable by an attacker who can observe the behaviour of a valid user. Because authentication is on the basis of responses to random challenges, a well-designed activity and feature set makes it hard for a software robot to succeed in authentication.

2. DRAW A CIRCLE (DAC)

DAC is a challenge-response behavioural authentication

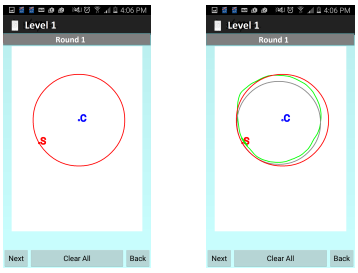


Figure 1: A challenge (red color) is a circle with a specific starting point. The response (green color) is the user drawing the circle staring from the givenpoint.

Feature	Domain	Notes
`Errors in drawing` $f_1 - f_{10}$	$0 \leq d_e \leq 200$	drawing error in 10 distinct points
`Error in starting point` f_{11}	$0 \leq d_s \leq 150$	error based on suggestion
`Direction of drawing` f_{12}	$(+1, -1)$	clockwise and anticlockwise direction
`Drawing time` f_{13}	$0 \leq t_d \leq 5$	time (normalize) to draw a circle
`Error in radius` f_{14}	$0 \leq d_r \leq 200$	radius difference based on suggestion
`Error in center` f_{15}	$0 \leq d_c \leq 150$	deviation of center
`Touch pressure` $f_{16} - f_{25}$	$0 < p \leq 1$	touch pressure in 10 distinct points
`Touch size` $f_{26} - f_{35}$	$0 < s < 1$	touch size in 10 distinct points
`Drawing speed` $f_{36} - f_{55}$	$0 < (v_x, v_y) \leq 4000$	touch speed in 10 distinct points
`Reaction time` f_{56}	$0 \leq t_r \leq 5$	time to start drawing after suggestion
`Miss count` f_{57}	$(0, 1)$	constrains to restrict random drawing
All features are part of both levels (exception f_{56}) and real number (exception f_{12}, f_{57}) All *error* is in pixel distance, *time* unit is in second and *speed* unit is #pixel/second		

Table 1: The set of features in DAC

system that uses the behaviour of users when drawing a circle, for authentication. The intuition here is that users' behaviour in complex activities are shaped by their intrinsic personal traits and this can be used for verifying their authentication claims. In DAC, a challenge is in the form of a circle with some constraints that the user is expected to draw. The user drawn response is a circle that is used to generate a vector of feature values. DAC system has a device and a server part, consisting of three modules: i) an *Interaction Module* that manages the touchscreen of the mobile device used by the user; ii) a *Data Acquisition Module* that runs on the mobile device and captures the user responses to send it to the server, and iii) *Verification Module* that is run by the server and matches the received data against the claimed identity profile and produces an *accept* or *reject* decision. The server software also includes challenge generation and transmission software. We assume that the communication between client and server is secured by SSL (Secure Socket Layer).

Game Design: The design of DAC, its feature selection and the matching algorithm was over a period of four months, working with a small focus group of 6 volunteers. We started with a four level game with 25 features in each level. In all levels, the user response was a circle based on random challenge. Through a series of experiments, we concluded that Level 3 and Level 4 produced similar results and so consolidated them into one, and Level 1 that did not sufficiently distinguish users was discarded. The final game has two levels. In Level 1, the challenge is a random circle with a randomly generated specific starting point that is presented to the user, and in Level 2, the challenge is a circle that is shown to the user for 3 seconds and then disappears. Figure 1 shows a challenge and response in DAC Level 1 and the corresponding user's response.

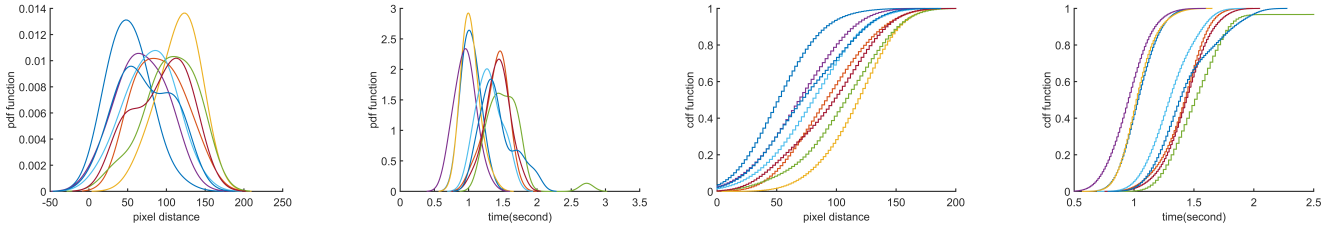
Feature Selection: DAC uses 57 features in each level. The collected features are, (i) a vector of user errors, each component corresponding to the discrepancy between challenge circle and an ideal circle drawn from the user drawing. (ii) features related to their physical characteristics and drawing skills, and iii) short-term memory ability. All features are selected through extensive experiments and shows in Table 1. We required the following properties for the features: i) *distinctiveness* that allows to distinguish users, ii) *stability (permanence)* that is needed for consistent measurement, iii) *non-delegation* which is to provide security against passing credential to others, and iv) *non-inferable cognitive ability*, not to allow feature values to be inferred by observation. The features from the same group (e.g. error in drawing), when carefully spaced, do not show correlation, and so in our matching algorithm, we treat them as independent features.

Feature Properties: Features in DAC capture different aspects of user behaviour and characteristics. Features f_1-f_{10}, f_{11}, f_{14}, f_{15} and f_{57} are related to user skill. Feature f_{12}, f_{13}, f_{16}-f_{25}, f_{26}-f_{35}, f_{36}-f_{55} capture intrinsic user properties. Feature, *touch size* (f_{26}-f_{35}), and *touch pressure* (f_{16}-f_{25}) are related to physical characteristics of the user and are stable and non-inferable features. Features f_1-f_{10}, f_{14}, f_{15} and f_{13}, f_{36}-f_{55} are rival features [1] and will not be easy to delegate as a pair (i.e. if the attacker wants to make one the features "closer" to the corresponding user feature, the second feature will further grow apart.) Time constraint f_{13} and miss-count f_{57} reduce the adversaries' success probability. Reaction time f_{56} is correlated to the short term memorization capability. Figure 2 shows the distribution (*pdf* and *edf*) of 2 features of 8 users. We used similar graphs to verify the suitability of other features. All features do not have the same level of user distinguishing capability. We rank them into 7 different groups and assign different weights for calculation of the final score. The rank of a feature is calculated by finding the contribution of the feature to the final authentication score. Table 2 shows the ranks of all features.

Verification Algorithm: Each feature in the profile is stored as a set of samples from a probability distribution. During authentication, a fresh vector of samples is generated. The matching algorithm uses two sample KS-test (Kolmogorov-Smirnov test) [4] (samples coming from the registration and the verification data) for each feature and produces a distance as well as p value. The matching algorithm combines the P_i values of the features and their calculated weights, to generate a combined P value that is used for authentication decision.

Rank	Level 1	Level 2
1	f_{14}	f_{14}
2	f_{15}	f_{15}
3	f_{11}	f_1-f_3
4	f_1-f_3	f_{11},f_{56}
5	f_8-f_{10}	f_8-f_{10}
6	f_4-f_7	f_{16}-f_{55}
7	f_{16}-f_{55},f_{12},f_{13}	f_{12},f_{13}

Table 2: Rank of all features



(a) Error in starting point (b) Drawing time (c) Error in starting point (d) Drawing time

Figure 2: The smooth **pdf** and **edf** of 2 features of 8 users. The graphs show that features can be used to distinguish users.

.

3. EXPERIMENTS

We applied and received *Ethics Approval* from the Research Ethics Board of our institution. We collected data in, i) a *controlled setting* where 6 graduate students participated in the design of DAC, and ii) an *uncontrolled setting* where 50 remote users participated in our experiments. We uploaded our app and user instruction to Google Play Store. During the registration phase, a user participated in 50 rounds of challenge and response for each game level. We considered the first 20 rounds as a practice rounds and the next 30 rounds are for profile generation (DAC also has a separate practice mode to allow users to practice if they desire to do so). We excluded 11% of the profiles as were outliers, possibly result of careless playing. In the authentication phase, we collected data from 20 challenge and response rounds. This number can be reduced by using more than one circle in one screen.

User Distinguishability of Profiles: A profile is represented by a set of n vectors of m feature values (each column corresponding to the samples of a feature). We measured the similarity between the profiles of two users based on the similarity of their features data and using KS-test. Our experiments show that on average two profiles overlap in 20% of features and in the worst case the overlap is around 33%. Here overlap is measured as the percentage of features that are considered similar by the KS-test. The overlapping features, however, depend on the profile pairs and varies for different pairs.

Correctness: We recorded 74 authentication attempts from 50 different users (some users with more than one attempt recorded at different times). For the threshold value τ=50 DAC has 94.10% of success rate with 5.40% false acceptance(FA) and 5.93% false rejection(FR) rate. If we increase the threshold to 55, success rate becomes 96.82% with 3.09% FA rate and still has the tolerable FR rate 6.75%.

Non-delegatability: In delegation attack, a user allows a third party to learn their behaviour by observing their behaviour during the response. A similar (weaker) learning attack happens in shoulder-surfing. To evaluate this type of attack, we videotaped 3 complete authentication attempts of 3 local users and then asked 8 users (2 local, but different, users) to emulate the behaviour of the 3 users. We did not put any restriction on the number of attempts. In most cases, the distribution of an imposter's feature had different mean and variance compared to the original data. Among the 6 imposters, the success rate of two imposters with the highest number of tries, 52 and 39 tries, respectively, was almost zero for the threshold τ=55. With threshold τ=50, the imposter with 52 attempts had 5 successes.

Stability of Features: Users who tried more than once had the success rate 92.34% with 6.12% failing in their first attempt. Motivated by this result we evaluated the stability of features by collecting user data at different times. Our results show that most features are stable and the distribution has negligible change over time. One can, however, use adaptive algorithms to vary threshold using user history data, to further eliminate the effect of this variation. This will be our future work.

4. CONCLUSION

DAC follows the framework of using user behaviour in well-designed activities for authentication. DAC is a simple and intuitive activity that generates a rich set of features that are distinguishing and non-delegatable. Drawing A Circle allows us to construct a vector of errors, that together with other physical and intrinsic features allow high accuracy for authentication. We are planning larger experiments to validate our results for wider populations. Another important future work is reducing the number of challenge and response rounds in an authentication attempt to improve the usability of the system. A straightforward way of achieving this is by using a challenge that consists of multiple circles. Finally using richer sensor data such as those for touch pressure and touch size will increase the accuracy of authentications.

Acknowledgement. This work is in part supported by Telus Communications Inc.

5. REFERENCES

[1] M. Alimomeni and R. Safavi-Naini. How *to Prevent* to delegate authentication. In *International Conference on Security and Privacy in Communication Systems*, volume 164, pages 477–499. Springer, Jan 2015.

[2] J. Bonneau, E. W. Felten, P. Mittal, and A. Narayanan. Privacy concerns of implicit secondary factors for web authentication. In *SOUPS Workshop on "Who are you?!": Adventures in Authentication*, 2014.

[3] H. Khan, A. Atwater, and U. Hengartner. Itus: an implicit authentication framework for android. In *Proc. of the 20th annual international conference on Mobile computing and networking*, pages 507–518. ACM, 2014.

[4] F. J. Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253):68–78, 1951.

[5] E. Shi, Y. Niu, M. Jakobsson, and R. Chow. Implicit authentication through learning user behavior. In *International Conference on Information Security*, pages 99–113. Springer, 2010.