Poster: A Continuous and Noninvasive User Authentication System for Google Glass

Ge Peng*, David T. Nguyen*, Gang Zhou*, Shuangquan Wang†
*College of William and Mary, USA, †Institute of Computing Technology, Chinese Academy of Sciences, China
{gpeng, dnguyen, gzhou}@cs.wm.edu, wangshuangquan@ict.ac.cn

Categories and Subject Descriptors

D.4 [**Security and Protection**]: Authentication

Keywords

Continuous authentication; Google Glass

1. INTRODUCTION

Google Glass stores many kinds of user data, including contacts information, messages and emails, photos and videos, and much more. This brings privacy risks to the owner. To protect user privacy, Google Glass uses four touch gestures selected by the owner from eight pre-defined gestures to authenticate users. This is invasive and also susceptible to peeking. In this work, we propose a continuous and noninvasive authentication system for Google Glass, named GlassGuard. GlassGuard discriminates the owner and an imposter with touch behavioral biometrics and voice features, which are available during a user's normal daily usage. Touch behavioral biometrics have been studied on smartphones. However, due to user interaction differences between Google Glass and smartphones (e.g. holding smartphones with hand(s) but wearing Google Glass on head, different gestures), systems proposed for smartphones cannot be applied directly on Google Glass.

2. DESIGN AND EVALUATION

Figure 1 shows the system architecture. Event Monitor continuously monitors user input events when the screen is on. The event data is then forwarded for feature extraction. For a touch event, a set of features are extracted from both touch data and sensor data. For a voice command, MFCC vectors are extracted. After features are extracted from a user event, they are passed to one of the classifiers depending on the event type. There are 7 classifiers in the system, each for a specific type of user event (T-Classifier for single tap, SF-Classifier for swipe forward, SB-Classifier for swipe backward, SD-Classifier for swipe down, TFSF-Classifier for two-finger swipe forward, TFSB-Classifier for two-finger swipe forward, and VC-Classifier for voice command). All classifiers are trained with one-class SVM model and they make predictions independently. The Aggregator employs a mechanism adapted from Threshold Random Walking [1] to combine multiple classification results and make the final decision when and only when it is confident. The Power Control module decides when to pause or restart data processing (feature extraction and classification) and sensor sampling according to the current privacy risk level.

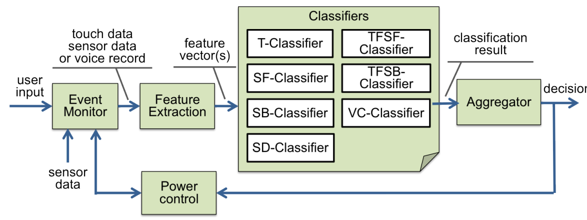

Figure 1: System architecture

We carry out a user study and collect real user interaction data from 32 subjects. With the data collected, we evaluate the performance of the classifiers as well as the whole system. Figure 2 shows the mean and standard deviation of Equal Error Rate for each classifiers in the system. Two-finger touch gestures provide lower EERs than one-finger touch gestures and VC-Classifier has the lowest EER 4.88%.

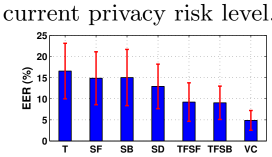

Figure 2: Classifiers

Table 1 shows the average performance of the system under five typical usage scenarios: 1) skim through the timeline. 2) delete a picture in the timeline. 3) take a picture and share it using voice commands. 4) take a picture and share it using touch gestures. 5) Google search. The decision delay is the number of events needed for the system to make a decision. Scenario 1 has the worst performance because it only consists of swipe forward gestures. Scenario 3 has voice commands. So, it has the best performance. Under all scenarios, the system has detection rate >93% and false alarm rate <3% after less than 5 user events.

Table 1: system performance

No.	detection rate	false alarm rate	decision delay
(1)	93.3%	2.84%	4.27
(2)	96.7%	1.42%	4.66
(3)	99.6%	0.24%	2.25
(4)	97.4%	0.87%	4.73
(5)	98.1%	1.08%	4.63

3. REFERENCES

[1] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan. Fast portscan detection using sequential hypothesis testing. In *Security and Privacy*. IEEE, 2004.