“Typing” passwords with voice recognition: How to authenticate to Google Glass

Daniel V. Bailey
Horst Görtz Institute for IT-Security
Bochum, Germany
danbailey@sth.rub.de

Markus Dürmuth
Horst Görtz Institute for IT-Security
Bochum, Germany
markus.duermuth@rub.de

Christof Paar
Horst Görtz Institute for IT-Security
Bochum, Germany
christof.paar@rub.de

ABSTRACT

Augmented-reality glasses like Google Glass present a new set of user-interface trade-offs which must be carefully considered in crafting user authentication protocols. First, it lacks a keyboard or touchscreen; second, the most prominent input mechanisms, voice recognition and a swipe sensor, are both easily observable by bystanders and thus are not suitable for password entry. Fortunately, these devices offer a private display that cannot easily be viewed by bystanders, and which helps in constructing secure user authentication. In this position paper we will outline problems and possible solutions for authentication on augmented-reality glasses.

1. INTRODUCTION

Wearable computing devices dispense with the keyboard in favor of a more radical rethinking of user interface. The new interfaces will present new trade-offs for usability and security, requiring a reinvestigation of even the most basic of protocols applied to the new settings. In particular, devices can be seen as offering input and output channels whose configuration gives them more or less protection from attackers. Focusing on augmented-reality glasses, this position paper discusses how we believe this development will influence user authentication for those devices, and in particular we argue for a shift away from traditional passwords.

Augmented-Reality Glasses.

A proposal which is still in an embryonic state is what we call augmented-reality (AR) glasses, with the most prominent example being Google Glass [2]. Due to their novelty, they offer very interesting research perspectives. AR glasses are worn on the head similar to eyeglasses: they can in fact be mounted on a pair of eyeglass frames. The form factor of those devices means that the available input methods are very different from PCs, tablets, or smartphones. In particular they do not offer keyboards or touch-screens that emulate keyboards. Taking the example of Glass, the following input methods are available (see, e.g., [1]).

- Textual input is typically provided using *speech recognition.*
- The side of the device is touch-sensitive (which requires the user to raise her arm to her head and thus is not ideal for long-term operation). Some *touch gestures* that are supported are swiping, tapping, and pressing, with one or more fingers.
- Glass recognizes eye gestures (winking and blinking), and the front-facing camera can recognize QR codes and printed text
- It offers access to built-in *gyroscope*, *accelerometer*, and *magnetometer*.

2. CHANNEL MODEL

The design and placement of the display means that contents should be visible only to the wearer: the pixels themselves are 1/8th the size of those found on iPhone 5, with a fixed-focus lens making the display appear to the wearer to be a few meters away. This quality reduces the likelihood of “shoulder surfing,” where a nearby attacker looks at the display. We will use this assumption below in crafting authentication protocols. Our reference setting is that of a wearer authenticating themselves to their AR glasses. The protocols should remain secure even from an attacker who can passively observe many instances before attempting to authenticate.

Challenge-response protocols have been treated extensively in the literature. A few factors combine to make this setting unique.

- A unidirectional secret and authentic channel from the device to the user.
- An insecure/observable back-channel from the user to the device.
- The protocol steps on the user-side need to be human-computable.

We aim to prove possession of a secret without leaking it to an observer, a task typically handled by a cryptographic protocol. Unfortunately, the secret, authentic channel is provided from the AR glasses directly to the human user. We must therefore rely on functions that can be computed by the wearer.

3. EXAMPLES

In the following, we discuss several examples of well-known user authentication schemes with respect to their usability on AR glasses, and demonstrate how the channel model introduced above can lead to secure input methods for some of them.

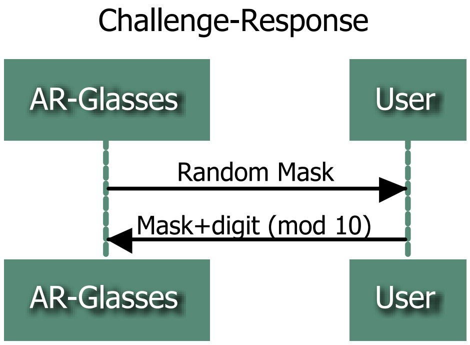

Figure 1: Simple challenge-response protocol.

Random PIN numbers.

PIN numbers chosen uniformly at random are an example for an authentication mechanism that can be made quite secure on AR glasses. Based on the channel model described above, one can use a simple challenge-response protocol to construct an authentication scheme provably secure despite an eavesdropper on the public channel (going from the user to the device).

Suppose the user has memorized a secret random PIN. We cannot simply have the user speak the digits, as anyone within earshot would learn the secret value. Instead, we can proceed on a digit by digit basis with the display showing a random digit as a challenge. In response, the user can add her secret digit and speak the result mod 10. The process is completed for each digit in the PIN, which a fresh random digit displayed each time (see Figure 1).

A variation of this theme, which does not require any computations by humans, uses the private channel (from the device to the human) in a more intrinsic way: for each individual digit, it displays a random assignments of the digits $\{0, \ldots, 9\}$ to another set (for e.g., $\{A, \ldots, K\}$), and the user reads out the associated letter.

Both schemes are provably secure against an eavesdropper, and should have acceptable usability. (The iterative character of the protocol certainly lowers usability, but, as each digit has "full entropy", a small number of rounds is sufficient.)

Textual passwords.

In principle, the same mechanisms can be used for textual passwords and user-chosen PINs. The drawback is, however, that they have a much lower entropy per character (Shannon gave estimates for English language of 1.1 bits per letter [4]), which increases the number of rounds for a straight-forward implementation and reduces usability.

Graphical passwords.

Recall-based graphical passwords schemes, such as the classical Draw-a-Secret (DAS) [3], or the recently deployed Android graphical password scheme, seem largely unsuited. Reasons are that they require fine-grained input, and that they have quite low entropy per character/token [5]. Most *cued-recall based schemes* require, in one form or another, to select points or other gestures on an image background, the classical example being PassPoints [6], which also forms the basis for the Windows 8 Picture Password scheme. For similar reasons, we do not see potential for those schemes to be implemented AR glasses.

Recognition-based schemes typically require that a user identifies previously seen images from a set of decoy images. Such a scheme can easily be realized by displaying randomly permuted labels for the images, which then can be spoken out loud, very similar to the second method for PIN entry discussed before. Alternatively, "blink" or "tap" gestures, or even head movements detected by the acceleration sensors, can be used. While recognition-based schemes are typically quite vulnerable to shoulder surfing attacks, they seem very well suited for AR glasses and similar devices.

Biometric systems.

Biometric schemes seem problematic, as they require extra hardware to be secure, and extra hardware adds to weight, which we do not expect to happen. The built-in camera cannot easily be used for face recognition, as it is facing away from the user.

4. CONCLUSION

Augmented-reality glasses, and wearable IT in general, require new techniques for user authentication. In this position paper, we explained some specifics of authentication on AR glasses and explain which known authentication schemes can be used, with little modifications, on augmented reality glasses.

5. REFERENCES

[1] Dapper Vision Inc. Wearscript, May 2014. `http://www.wearscript.com/en/latest/input.html`.

[2] Google Inc. The Glass Explorer Program, May 2014. `http://www.google.com/glass/start/`.

[3] A. D. Rubin, I. Jermyn, A. Mayer, F. Monrose, and M. K. Reiter. The design and analysis of graphical passwords. In *8th USENIX Security Symposium*, pages 1–14. IEEE, 1999.

[4] C. Shannon. Prediction and entropy of printed english. *Bell Systems Technical Journal*, pages 50–64, 1951.

[5] S. Uellenbeck, M. Dürmuth, C. Wolf, and T. Holz. Quantifying the security of graphical passwords: the case of android unlock patterns. In *ACM Conference on Computer and Communications Security*, pages 161–172, 2013.

[6] S. Wiedenbeck, J. Waters, J.-C. Birget, A. Brodskiy, and N. Memon. Passpoints: Design and longitudinal evaluation of a graphical password system. *Int. J. Hum.-Comput. Stud.*, 63(1-2):102–127, July 2005.