PPG²Live: Using dual PPG for active authentication and liveness detection.

Jan Spooren

imec-DistriNet, KU Leuven
Celestijnenlaan 200A, 3001 Heverlee, Belgium
jan.spooren@cs.kuleuven.be

Davy Preuveneers

imec-DistriNet, KU Leuven
Celestijnenlaan 200A, 3001 Heverlee, Belgium
davy.preuveneers@cs.kuleuven.be

Wouter Joosen

imec-DistriNet, KU Leuven
Celestijnenlaan 200A, 3001 Heverlee, Belgium
wouter.joosen@cs.kuleuven.be

Abstract

This paper presents a novel solution based on PPG to strengthen face authentication. Our method leverages and combines different PPG signals from multiple channels to meet two objectives. First, it complements face authentication with an additional authentication factor, and second, it strengthens the liveness detection to be more resistant against presentation attacks. Our solution can be implemented as an unlock screen for mobile phones having a front-side and back-side camera or a paired smart watch as well as for webcam-enabled laptops augmented with a PPG sensor. Our evaluation shows that our method can significantly improve the resilience against presentation attacks for face recognition-based user authentication.

1. Introduction

User authentication which is both reliable and unobtrusive is still a poorly solved problem. Passwords are known to be problematic: they are cumbersome to remember and to type, often easy to guess, and can be shared or leaked. Hardware tokens can be costly, a burden to roll out, not supported by all environments and some require a numeric code to be typed, which is even more inconvenient than typing a password known to the user. They are also not very suitable for unlocking mobile devices.

Biometrics can provide a solution, provided that the proposed approach is resistant against *presentation attacks* [10] (also known as *spoofing*), whereby an adversary provides similar-looking biometric features in order to spoof the authentication mechanism. Particularly for face recognition,

many presentation attacks exist, ranging from printed photographs, photographs with eyes cut out through which the adversary looks in order to thwart eye-blinking detection, to recorded video of the person the adversary tries to impersonate. A recent study [7] of January 2019 showed that 42 out of 110 tested smartphones are vulnerable to this attack. Since vast amounts of images and even video footage of many people is readily available nowadays (such as on social networks), it is often trivial for adversaries to obtain facial images for mounting face recognition presentation attacks. The state of practice of detecting whether a face image or video belongs to a real person in front of the camera is often called *liveness detection* [10].

In this work, we provide a novel approach to liveness detection, based on capturing the heart rate through a photoplethysmography sensor, as well as deducing the same heart rate from the actual face video that is used for authentication through face recognition. This solution makes it harder for an adversary to spoof face authentication.

In section 2, we discuss the relevant related work, followed by an overview of the taken approach in section 3, where we discuss the envisioned authentication scenarios and the studied techniques for matching a PPG and rPPG signal in the time and frequency domain. We then present our method of evaluation in section 4, followed by a discussion of the results in section 5 and present our conclusions in section 6.

2. Related work

Many techniques for liveness detection have been proposed with varying levels of robustness, which can be roughly classified into the following categories:

- **Image quality analysis:** Printed images tend to have less contrast and a more limited color spectrum than

real objects. Examining picture quality can therefore be a feature for spoofing detection. In its simplest form, *color moments*, such as the mean, standard deviation and skewness of the RGB or YUV color channels of an image or video can be used to distinguish real recordings from spoofed ones.

- **Texture based:** Ojala et al. [12] proposed a technique called *Local Binary Patterns* (LBP) as a powerful feature for texture classification. It has since been used and extended extensively in multiple variants by many different authors [2, 14, 18, 4]. LBP can distinguish the texture from a photograph or a tablet screen, compared to a real human in front of a camera and is therefore frequently used as a feature for liveness detection. Rahtu et al. [13] discovered a blur-insensitive texture characterization method, called *Local Phase Quantization* (LPQ).
- **Motion based:** Kollreider et al. [6] used a technique called *Optical Flow of Lines* and Gabor feature extraction to capture subtle movements in the face, assuming that particular spots on a real face move differently to that same face presented on a photograph. Chaudhry et al. [3] proposed *Histograms Of Optical Flow* (HOOF) as a technique, which can similarly be used to differentiate between a moving photograph and a real person on video.
- **Blood flow based:** Blood absorbs light in the green spectrum. Although this is unnoticeable to the human eye, the variation of the green color channel due to pulsating blood vessels in the face can be detected in video taken at the standard color depth of 8 bits in R, G and B. Typically, this requires a bandpass filter between 0.5 and 5 Hz (the human pulse frequency range) and an FFT transform. Since the peaks detected are weak and often drowning in noise, it helps to compare different regions in the face with regions in the background to detect peaks measured only in the face and absent in the background. This technique is called *Remote Photoplethysmography* (rPPG).

Wang et al [15] use three sets of features and an SVM classifier to detect liveness. They use (M)LBP, block-based color movement and a new feature *Red-Green Deviated Texture*, which consists of the absolute value of the LBP histograms of the green and red channels subtracted, which is claimed to detect effects of blood flow in the face. The technique works on still images and hence the term '*Blood Flow Analysis*' in the title could be considered somewhat misleading. They claim good results on the NUAA, CASIA, Idiap and MSU spoofing datasets, however, these datasets can be considered outdated with regards to the current state of the art in mobile camera resolution and picture quality.

Nowara et al. [11] proposed the use of photoplethysmography to detect a live person. They used the color varia-

tions in the green channel on a person's forehead and left and right cheek to detect subtle variations in the 0.5 to 5 Hz range, which were present in all three facial locations, but not in the background to prove the liveness of the person filmed. However, we believe that this technique can be spoofed: by illuminating a photograph or a facial mask of a person who is being impersonated with a light source in which the green levels have been made to oscillate at a typical heart rate frequency, this technique can be fooled to detect liveness. Illuminating a photograph or mask with such a light source can be as simple as playing back a specially crafted movie clip on a smart phone near the subject's spoofed face.

Atoum et al. [1] use two convolutional neural nets (CNN) to try to detect spoofed images: one is fed with local binary patterns on random spots of individual frames, the other tries to extract depth information from the pictures: It is trained with a flat depth map for known spoofed images and with an artificially generated depth map, deduced from the facial geometry for known-authentic images. During testing, the latter CNN will output a flat or a 3D-type of depth prediction. This output is then used to predict spoofed versus authentic faces. It is not clear if the depth estimation actually contributes to the prediction, or if the depth-based CNN is trained to generate flat depth maps, when it detects artifacts that indicate a spoofed image, such as fingers holding up a tablet (as is the case in the low-resolution version of the CASIA-FASD [17] dataset). The predictions for the two CNNs are then fused by means of a weighted average of the scores of the two CNNs.

Liu et al. [9] continued on [1] but additionally include a Long Short-Term Memory (LSTM) [5] neural net to predict an rPPG signal as an additional liveness check. The approach is novel in that a recurrent neural net is used to extract an rPPG signal from video footage. However, it is difficult to ascertain that the rPPG signal is actually present in the images, or that the LSTM created an rPPG-alike signal to indicate that it predicts the video not to be spoofed. This is illustrated by the images in fig. 8(b) of their paper, which shows a spoofed image with a realistic looking rPPG spectrum. Moreover, the fusion algorithm used is the sum of the predicted rPPG FFT spectrum average and the predicted image depth average. Therefore, this method could be vulnerable to the aforementioned technique of illuminating a print with a heart rate like signal, e.g. from a smartphone screen. This could be an example of the failure of the multi-factor paradigm, where spoofing one easily-spoofed factor can in fact jeopardize the entire classification, rather than the multiple factors making the classifier stronger. However, this would have to be verified with the actual implementation.

Yadav et al. [16] used a fingertip-collected PPG signal by itself as an authentication mechanism, using *Continuous*

Wavelet Transform (CWT), followed by *Direct Linear Discriminant Analysis* (DLDA) for dimensionality reduction and using the Pearson distance as a classifier for template matching. While this is an interesting technique, it appears that the current state of practise does not allow this technique to be used as a sole authentication factor: The equal error rates (EER) are still high when there is a time gap between the training (i.e., the 'enrollment' of the user) and the test. Moreover, the experiments conducted are based on a small number of subjects (30 to 40) and no separate fold of the data was set apart for parameter tuning, which leads us to believe that the published EER rates may be optimistic, compared to a real deployment as an authentication mechanism.

Kurylyak et al. [8] showed that a smart-phone can easily be used as a photoplethysmography sensor, when the subject covers the camera with her fingertip.

The gap with the state of the art which we address in this work is the fact that although some prior work has tried to use remote photoplethysmography as a liveness check, we believe that the current state of practice suffers from the fact that it can easily be spoofed and is therefore to be considered as a very weak form of liveness check. Moreover, we believe that using two factor authentication where both factors can be linked with a common heart rate signal from the authenticated subject, can make spoofing either of these factors much more difficult and can therefore significantly strengthen the authentication mechanism against presentation attacks.

3. Approach

In this work, we introduce a novel technique of multifactor authentication, which we call *PPG²Live*, where the two used factors are linked, resulting in a considerably stronger form of authentication than the individual factors, separately.

As a first factor, we use face recognition, with as liveness check, the extraction of an rPPG signal from the facial video. As argued in section 2, extraction of an rPPG signal from facial video in itself does not constitute a strong liveness check, since this signal can easily be added to a facial mask or facial image, by illuminating the mask or image with a PPG-alike signal.

As a second factor we use a PPG signal, extracted e.g. from a smart watch, a smart phone camera [8], a pulse sensor or fingerprint sensor. As shown in [16], a PPG signal by itself can be used for authentication purposes albeit with relatively high Equal Error Rates (EER).

The strength of the approach is that the two factors are strongly linked by means of the PPG signal, in both factors generated with the subject's heart as the original signal source. This makes spoofing the individual factors separately considerably more difficult.

Figure 1. PPG signal collected from a finger tip sensor (blue signal, detected peaks marked with circle), recorded simultaneously with rPPG signal from the green channel of a laptop web cam (red signal, diamond). Signals normalized to the interval [0.0, 1.0].

3.1. Authentication scenarios

Although acquiring the subject's heart rate for authentication purposes may seem cumbersome, it can actually be done relatively simply. We see several different possibilities:

1. Authenticating to a laptop

(a) Using a smartwatch and the laptop's webcam: The subject wears a smart watch paired to a laptop or alternatively, paired to a smart phone which can communicate with the laptop. The smart watch records the user's pulse through its built-in PPG sensor. The watch in itself is a '*something-you-have*' authentication factor, through the fact that it is paired, either to the laptop or the smart phone. The PPG signal from the smart watch can be correlated to the rPPG signal extracted from the laptop's web cam which uses face recognition (constituting a '*something-you-are*' authentication factor).

(b) Using a fingerprint reader with built-in PPG sensor: The fingerprint is a '*something-you-are*' authentication factor. Similarly to face recognition, fingerprint readers must be made resistant to presentation attacks, using a liveness check. A PPG sensor in the fingerprint reader can be used as a fingerprint liveness check. This PPG signal check can be correlated to the rPPG signal extracted from the laptop's webcam, which is used as another '*something-you-are*' authentication factor.

Figure 1 shows the signals of a PPG sensor and the simultaneously extracted rPPG signal from a webcam.

2. Authentication to a smart phone

- (a) Similar to the authentication to a laptop, a smart phone can be paired to a smart watch. The presence of the smart watch is a ‘*something-you-have*’ authentication factor. The smart phone’s selfie camera is a ‘*something-you-are*’ authentication factor and both can be linked through the heart rate sampled by the smart watch PPG sensor as well as deduced from the rPPG signal extracted from the selfie-cam video.
- (b) Most smart phones are equipped with at least two cameras: One front-side camera, used as a ‘selfie cam’ and one back-side camera, used for general-purpose photography. The selfie cam can be used for authentication using face recognition, while the subject covers the general-purpose camera with her finger tip [8]. Both cameras return a PPG signal. The general-purpose camera’s PPG signal is of good quality and can possibly be used for authentication [16]. From the selfie cam’s facial authentication video, an rPPG signal can also be extracted and both can be correlated for stronger authentication, guaranteeing that the face used for authentication, belongs to same person as the finger used for PPG authentication.
- (c) Similar to the authentication to a laptop, smart phones often feature fingerprint readers, which can be used to provide a fingerprint, as well as a PPG signal, which can be compared to the rPPG signal from the selfie camera.

3.2. Techniques to match the PPG and rPPG signals

A number of techniques can be used to establish that the heart rate measured through the PPG sensor and the rPPG signal extracted from the subject’s facial video are both signals originating from the same source.

3.2.1 Time domain

Comparing the PPG and rPPG signals in the time domain is somewhat hindered by the fact that the rPPG signal extracted from the subject’s facial video is typically much more noisy and tends to be drifting, due to changing lighting conditions and natural movements of the subject’s face. Therefore, comparing the PPG and rPPG wave forms using convolution or correlation did not prove to be a practical classification feature.

A band pass filter between 0.5 Hz and 5 Hz could be used to remove some of this noise. However, bandpass filters tend to exhibit a settling time, which makes it difficult to rapidly match the two heart rate signals.

We therefore adopted the approach to compare the peaks in the PPG and rPPG signals, in order to determine if both signals originated from the same source: the subject’s heart.

As can be observed from Figure 1, the signal collected by a PPG sensor reveals very distinct systolic peaks as well as smaller diastolic peaks. To reliably detect the subject’s heart rate, first an approximate heart rate is deduced by finding the first (non-DC component) peak in the Fast Fourier Transform (FFT) of the PPG signal. This peak represents the ground frequency of the heart rate, but is not very accurate, given the FFT bin distance of f_s/N , with f_s the sampling frequency and N the number of samples taken. Therefore, when 10 s of data are captured at a sampling rate of 50 Hz*, the resolution in the frequency domain is 0.1 Hz, corresponding to 6 bpm.

The thus calculated approximate heart rate frequency f_{approx} is then used to perform peak detection on the PPG signal with the distance condition that the detected peaks should at least be $0.8f_s/f_{approx}$ samples apart. This ensure that only the very outspoken systolic peaks are detected and the other local maxima are ignored. The result of this procedure can be viewed as the blue, round markers shown in Figure 1.

The rPPG signal, on the other hand, shows less identifiable features (which can also be deduced from the absence of harmonic peaks in the FFT diagram, shown in Figure 2). Therefore, the most straightforward manner to establish that the two heart rate signals have the same heart rate as a source is by identifying the systolic peaks in both signals and analyzing the phase differences between those peaks. Delay between the peaks is natural and can both be explained by physical properties of the human body (such as a different vein diameter and distance from the heart at the two measured body locations), as well as by delays in capturing and transmitting the signals (e.g., the bluetooth communications channel between a smart watch and a smart phone). However, for a given configuration of sensors and body locations, the delay between both peaks is expected to be constant. Therefore, a set of classification features can be the average peak delay, as well as its standard deviation.

3.2.2 Frequency domain

Analysis in the frequency domain did not yield any useful classification features:

Fast Fourier Transform would be an evident technique to compare the PPG and rPPG signals in the frequency domain. The problem, however, (as mentioned in section 3.2.1) is the accuracy in the frequency domain: The distance between frequency bins in an FFT spectrum is too high to distinguish different subjects: a typical accuracy of 0.1 Hz (or 6 bpm) would imply that two subjects, one with heart rate 72 bpm and the other with heart rate 68 bpm, would be identified as the same subject.

*or any sampling rate, since the sampling rate disappears from the equation, given that $N = f_s t$

Figure 2. The FFT of the PPG and rPPG signals of figure 1

Moreover, as can be seen in Figure 2, although the (quite clean) finger PPG signal has typically up to 3 harmonics visible in the FFT spectrum, the much noisier rPPG signal only carries the heart beat ground frequency, yielding no elements for comparison other than the too inaccurate ground frequency.

Also, neither the use of Hilbert Huang Transforms nor Independent Component Analysis produced useful classification features.

4. Evaluation

We evaluated the *PPG²Live* technique by setting up an experiment, using a laptop webcam and an Arduino UNO, connected to a PPG pulse sensor. The Arduino was connected to the laptop, using serial communication over USB, at a speed of 57,600 bps. This setup allowed capturing video frames at a resolution of 800×450 pixels and frame rate of 55 fps, while simultaneously capturing PPG samples at a sampling frequency of 97 Hz. Both the video frames and PPG samples were timestamped immediately after capturing.

We used Python 3.5.2 with OpenCV 3.4.1 for image processing, NumPy 1.14.2 and SciPy 1.1.0 for math and signal processing, and dlib 19.4 for face detection.

The technique was tested on a group of 12 (8 male, 4 female) subjects aged between 22 to 51 years old, who were asked to have 10 seconds of their face recorded, together with their pulse, using the PPG sensor on one of their fingers. They were asked to keep face movements to a minimum.

To extract the rPPG signal from video frames, we used the *dlib* frontal face detector to find the face rectangle on the video frame half way the video footage and then took the average green value of this same rectangle over all of the recorded frames.

We then calculated the average time difference between corresponding systolic peaks in two (r)PPG traces, as well as the standard deviation on these time differences. We per-

Figure 3. The distribution of the two features extracted from the two PPG signals: the average time delay between corresponding systolic peaks detected in both signals, and the standard deviation on these time delays.

		Truth	
		Spoofed	Authentic
Prediction	Spoofed	110	1
	Authentic	1	11

Table 1. Confusion matrix for the classification, based on $stddev_delta > 0.064859$

formed this operation on each combination of rPPG and PPG traces for all recorded subjects, labelling the combination of two (r)PPG traces from different subjects and sessions as *spoofed*, and the combination of two traces taken during the same session, with the same subject as *authentic*. The result is shown in Figure 3.

Classification, solely based on the standard deviation on the systolic peaks in two traces results in near perfect classification with the rule that a session is spoofed if $stddev_delta > 0.064859$. Building a C4.5 (J48) tree, a PART decision list or using AdaBoost with Decision Stumps all result in the same confusion matrix show in Table 1, where out of 144 classifications, we record one False Positive and one False Negative classification.

5. Discussion

Face recognition is a well understood and mature technology; however, when used for authentication, resilience against presentation attacks needs to be extremely high and becomes an equally important consideration than the mere recognition of a subject. ‘Liveness checks’ are a mitigation against these presentation attacks, but can themselves be attacked, especially when their nature becomes known to adversaries.

A second authentication factor can be added instead, such as a fingerprint, PPG authentication or possession of a paired smart watch. A successful presentation attack would then need to successfully spoof both factors. With the technique we propose, those factors become linked together through the subject's heart beat as a common signal source. This offers a significant improvement to the use of rPPG as a liveness check, which is now much harder to spoof since only one particular heart rate signal will be accepted as a correct rPPG signal. Similarly, spoofing the second factor becomes equally harder, since also its liveness check is linked to the user's face authentication.

The proposed technique uses the timing of the systolic peaks in the two (r)PPG signals to establish if both signals are generated by the same source. This limits the use of the subject's PPG waveform to only a limited number of data points, rather than the full recorded curves. This is due to the noisy rPPG signal, which does not appear to allow better signal comparison. However, despite this limited use of the (r)PPG signals, the classification based on the standard deviation of the timing differences between the corresponding systolic peaks in the two (r)PPG signals offers a very useful technique for ensuring that both signals are originating from the same subject.

The proposed *PPG²Live* technique can be executed in real time during authentication and requires little or no extra hardware to use.

6. Conclusions

We are pleased to present *PPG²Live*, a novel technique to improve the strength of multi factor face authentication. By deducing a heart rate signal from the video used to perform face recognition on a subject, and comparing this heart rate signal to a heart rate signal obtained from a PPG sensor (such as in a smart watch or simply using a smart phone camera), we link the two authentication factors together and thus create a significantly stronger resilience against spoofing attacks.

The technique can be executed in real time and requires little or no extra hardware.

References

- [1] Y. Atoum, Y. Liu, A. Jourabloo, and X. Liu. Face anti-spoofing using patch and depth-based cnns. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 319–328, Oct 2017.
- [2] O. Barkan, J. Weill, L. Wolf, and H. Aronowitz. Fast high dimensional vector multiplication face recognition. In *2013 IEEE International Conference on Computer Vision*, pages 1960–1967, Dec 2013.
- [3] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1932–1939, June 2009.
- [4] M. Guermoui and M. Djamel. Modified local binary pattern for human face recognition based on sparse representation. *International Journal of Computers and Applications*, 36:64–71, 08 2014.
- [5] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(9):1735–1780, Nov. 1997.
- [6] K. Kollreider, H. Fronthaler, and J. Bigun. Non-intrusive liveness detection by face images. *Image and Vision Computing*, 27(3):233 – 244, 2009. Special Issue on Multimodal Biometrics.
- [7] P. Kulche. Gezichtsherkenning op smartphone niet altijd veilig, 2019 (accessed January 3, 2019). <https://www.consumentenbond.nl/veilig-internetten/gezichtsherkenning-te-hacken>.
- [8] Y. Kurylyak, F. Lamonaca, and D. Grimaldi. Smartphone based photoplethysmogram measurement. *Digital image and signal processing for measurement systems*, pages 135–164, 2012.
- [9] Y. Liu, A. Jourabloo, and X. Liu. Learning deep models for face anti-spoofing: Binary or auxiliary supervision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [10] S. Marcel, M. S. Nixon, and S. Z. Li. *Handbook of Biometric Anti-Spoofing*. Springer, London, 2014.
- [11] E. M. Nowara, A. Sabharwal, and A. Veeraraghavan. Ppgsecure: Biometric presentation attack detection using photoplethysmograms. In *2017 12th IEEE International Conference on Automatic Face Gesture Recognition (FG 2017)*, pages 56–62, May 2017.
- [12] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.
- [13] E. Rahtu, J. Heikkilä, V. Ojansivu, and T. Ahonen. Local phase quantization for blur-insensitive image analysis. *Image and Vision Computing*, 30(8):501 – 512, 2012. Special Section: Opinion Papers.
- [14] J. Trefny and J. Matas. Extended set of local binary patterns for rapid object detection. In *Proc. Computer Vision Winter Workshop*, pages 1–7, 2010.
- [15] S.-Y. Wang, S.-H. Yang, Y.-P. Chen, and J.-W. Huang. Face liveness detection based on skin blood flow analysis. *Symmetry*, 9(12), 2017.
- [16] U. Yadav, S. N. Abbas, and D. Hatzinakos. Evaluation of PPG biometrics for authentication in different states. *CoRR*, abs/1712.08583, 2017.
- [17] Z. Zhang, J. Yan, S. Liu, Z. Lei, D. Yi, and S. Z. Li. A face antispoofing database with diverse attacks. In *2012 5th IAPR International Conference on Biometrics (ICB)*, pages 26–31, March 2012.
- [18] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–928, June 2007.