Video-based Fingerphoto Recognition with Anti-spoofing Techniques with Smartphone Cameras

Chris Stein[1], Vincent Bouatou[2], Christoph Busch[1]

Hochschule Darmstadt - CASED[1]
Mornewegstraße 32
D-64295 Darmstadt

Safran Morpho - Identification Division[2]
Boulevard Gallieni 11
F-92130 Issy les Moulineaux

chris.stein@cased.de, vincent.bouatou@morpho.com, christoph.busch@h-da.de

Abstract: This work is concerned with the acquisition of fingerprints samples on smartphones with the built-in smartphone camera. A novel approach to capture multiple fingerphotos in a videostream with a smartphone camera and the processing of the photos for the finger recognition is discussed in this paper. The proposed technique offers a convenient and efficient way to capture multiple samples of a biometric instance in a short time frame. Due the fact that fingerphotos can be easily replicated with low effort (e.g. print outs with an ordinary printer) and thus are vulnerable to presentation attacks, anti-spoofing algorithms were developed to detect such spoof attempts. The algorithms for the detection and segmentation of the finger as well the preprocessing of the photo with graphical operations and anti-spoofing were implemented in a prototype as application for the Android operating system. User tests are performed to evaluate the usability and to create a database of biometric samples for offline evaluation of the recognition performance. Further tests are done with diverse artefacts such as printed finger images, fake fingers of gelatin, gummy and silicon as well finger replay videos to measure the resistance of the developed solution against presentation attacks.

1 Motivation

1.1 Field of application

Smartphones can be secured with a biometric authentication system based on fingerprints that use the fingerphoto recognition. The built-in camera of the smartphones is used to capture the biometric characteristics of the finger. The latest smartphones have at least one integrated camera to capture the finger in sufficient quality and have enough computational capacities to process the photos and execute algorithms for the fingerphoto recognition. Hence, there are no extra devices needed to perform the solution proposed in this work. Biometrics offers an authentication factor that is more reliable since knowledge-based authentication schemes since observed biometric characteristics cannot be delegated, forgotten or copied like e.g. passwords.

1.2 Advantages of the capture method

The proposed capture method with a camera has advantages over the widely spread used touch-based solutions for capturing fingerprints: Body contact is avoided while capturing the fingerprint sample and thereby there is no risk of leaving a latent fingerprint on a sensor. There are no deformations of the finger potentially caused by high pressure of the finger on a touch-based sensor and thus no risk of decreased quality of the captured sample due inadequate pressure. The video stream input enables the possibility to capture multiple samples from a biometric instance in a short time frame and with minimal user interaction. Such video frame sequences can be used to improve the quality of the biometric templates by consolidating the biometric information from multiple frames.

2 Related Work

This work is related to the work by Lee et al. [LD-2008], Derawi et al. [DM-2011] and the work of Stein et al. [SC-2012]. In the related work of Stein et al. a first complete authentication system for the Android operating system based on single fingerphotos was developed and evaluated. The prototype was written as a module for MBASSy [WH-2010]. MBASSy is a framework which allows the user to utilize various biometric authentication methods. The observed Equal Error Rate (EER) was in the range of 20% [SC-2012]. In that work a minutia extractor and comparator with low complexity was used to be executed directly on the smartphone. These components seem to be the main reason for the weak performance. In this work we have adopted the finger detection and segmentation algorithms from Stein et al. and have integrated an industrial solution for minutia extraction and comparison. Thus in this work, we are able to evaluate the recognition rate of a single photo capture and to benchmark it with our proposed video-based capture method that is based on an industrial minutia extractor and template comparator [Morp].

3 Objectives and Approach

The intention of our work was to improve the proposed finger authentication system for the Android smartphones [SC-2012] in terms of enhanced usability, recognition rate and anti-spoofing resistance. The capture method with single fingerphotos will be replaced by a video-based approach. The algorithms for the finger detection and quality assurance are adapted and optimized for continuously use on the video stream. An anti-spoofing technique is implemented that requires performing a challenge response of the user. The position and distance of the finger as well the edge density (metric for sharpness on fingerphotos [SC-2012]) and the light reflection on the finger caused from the LED of the camera is measured to detect spoof attempts. The developed solution is evaluated in user tests to determine the usability and to create a biometric database of fingerphotos that is used in an offline evaluation to determine the recognition rates. A commercial minutia extractor and template comparator namely the MorphoLite SDK from

Morpho [Morp] is applied to extract and compare minutiae in an offline technology test. The algorithms for spoof detection were tested with genuine presentations and different attack presentations with diverse artefacts such as printed finger images, fake fingers of gelatin, gummy and silicon as well finger replay videos.

4 Hardware Requirements and Camera Settings

In order to capture useable fingerphotos that contain the friction ridge pattern of the finger, the built-in smartphone camera must be able to focus on very close objects (<10cm) in front of the camera. The camera must also have a built-in LED that is used for the implemented anti-spoofing technique. The continuous capture and processing of the frames of the video stream demand a smartphone with at least a powerful dual-core CPU with more than 1GHz clock frequency in order to process the frames fast enough. Otherwise the finger and anti-spoofing detection rates can be affected negatively because too many frames cannot processed in time and must be discarded. Suitable smartphones those fulfill these requirements were the Galaxy Nexus and Galaxy S3 from Samsung.
The "macro" mode of the camera is used, such that the camera uses the closest possible focus. The LED is switched on during the capture process. The LED spotlights the finger such that it appears brighter than the background. This simplifies the detection and segmentation of the finger against the background. Another advantage is the reduced camera noise and risk of blurring caused from hand-motion due the high brightness from the LED. Further advantages using the LED are stabilized lighting conditions and a more homogeneous illumination. The usage of the LED is also important for the implemented anti-spoofing technique *(see Section 8)*.

5 Capture Process

The user simply positions his finger close in front of the camera *(see Figure 1)* in order to capture a defined amount of biometric samples from the video stream. The orientation of the finger can be random. During the capture the user has the option to rotate his finger slightly in x- and/or y-axis to capture the finger from different perspectives. The usage of multiple perspectives of the finger can improve the recognition rates when a consolidated template from several video frames is generated. The constant input stream from the camera is processed by the finger detection and quality assurance algorithms that is adapted for video stream input of the camera from the prior work of Stein et al. [SC-2012] to detect the Region of Interest (ROI) and determine the quality of the sample. The amount of the processed frames (fingerphotos) per second is limited by the processing power of the CPU. Frames that pass the quality assurance will be segmented, preprocessed and stored for the offline evaluation.

Fig. 1: Capture of the finger with the smartphone camera

6 Segmentation and Preprocessing of the Photos

The ROI of the captured photos that have passed the quality assurance will be further processed to prepare it for the minutiae extractor. These steps are applied on the ROI with the functions of the OpenCV framework [OSCV] in the following order:

1. Segmentation of foreground and background area
The foreground area (the finger) is segmented from the background to remove the pixels at the background that are not relevant for the fingerphoto recognition. This can be achieved when all values of the ROI below a defined value are set to black. All other values remain unchanged. This results in the segmented finger foreground area. Only the red channel is evaluated for the segmentation. A threshold of 100 (red value range 0…255) has been proven as an optimal value for the segmentation.

2. Transformation of the image from RGB to gray-scale
The color information is not used anymore after the ROI was detected from the finger detection algorithm. The computation of only one channel reduces the computation time for the following preprocessing steps significantly.

3. Median filter
A simple median filter with a kernel size of 3 is applied to reduce the camera noise.

4. Adaptive threshold
The ROI is binarized after this operation. The calculation is done by analyzing the gray values of the neighborhood pixels of a certain block size to determine the average value. A pixel is set to "white" if this average is above the threshold; otherwise it is set to "black". The Morpho minutiae extractor can handle with regular and inversed binarized images. Thus, an inversion of the binarized data (valleys are "white" and ridges are "black") would also work properly as input for the minutiae extractor. The binarization step is required for reliable detection of the minutiae of the finger with a minutiae extractor. The best results were achieved with a block size of 19 in combination with the used input resolution of 1280x720 pixels from the camera of the test device and the Morpho minutiae extractor.

5. Scaling to a fixed width
The dimensions of the images must be normalized because the capture method allows different distances of the finger to the camera those results in different dimensions of the image. The ROIs width is scaled to a fix value and the height is changed according the calculated scale factor to keep the aspect ratio. This operation ensures to generate from finger images always a geometrically normalized template that can be processed with the template comparator.

6. Cropping of the height
A very long image indicates that the border of the first finger segment was not properly detected. In this case the lower part of the image that does not contain any essential information for the fingerphoto recognition is removed, such that it does not exceed the defined maximal height.

Fig. 2: Preprocessing steps with the detected ROI from the finger detection algorithm: 1. Rotated ROI, 2. Segmentation and RGB-to-gray-scale transformation, 3. Median-filter and adaptive threshold, 4. Scaling to a fixed width and cropping height

7 Implementation

The application is written in Java for the Android operating system. Common middle to high end smartphones had at least a dual-core processor. The program workflow is optimized for dual-core processors to maximize the performance on such devices. The preprocessing of the fingerphotos requires more computing power than the finger detection and quality assurance algorithm together. Thus, the preprocessing is done in a separate (asynchronous) worker thread, so it does not block the main thread due heavy work load. This also allows the parallel preprocessing of photos and the capture of frames. The open source framework OpenCV [OSCV] is available for the Android operating system and used to perform the graphical operations on the images. For performance optimization, the preprocessing code is called over the JNI (Java Native Interface). The anti-spoofing algorithms are running also in separate thread to guarantee a high performance during the evaluation of the challenge response.

8 Presentation Attack Detection

8.1 Principle

After the probe photos are taken in authentication mode, the application enters the challenge response mode. In this mode, the user is prompted to move his fingertip slowly towards the camera. The shape and the consistency of the finger and in combination with the slow movement of the fingertip towards the camera lead to a characteristic strong reflection at the fingertip from the cameras LED. Other materials like 2D print outs and (unprocessed) fake fingers do not possess such reflection properties and thus do not pass the challenge response. The reflection must be detected near the fingertip and must be strong enough in order to exceed the defined threshold for a positive challenge response. *Figure 3* shows the reflection characteristics of a genuine finger and other typical fake fingers. The calculation of the light reflection is described in *Section 8.3*.

Additional checks regarding the edge density and the position of the finger as well the distance of the finger to the camera are performed to detect unusual sharpness values and keep the link with the shown finger from authentication mode *(see next Sub-Section 8.2).*

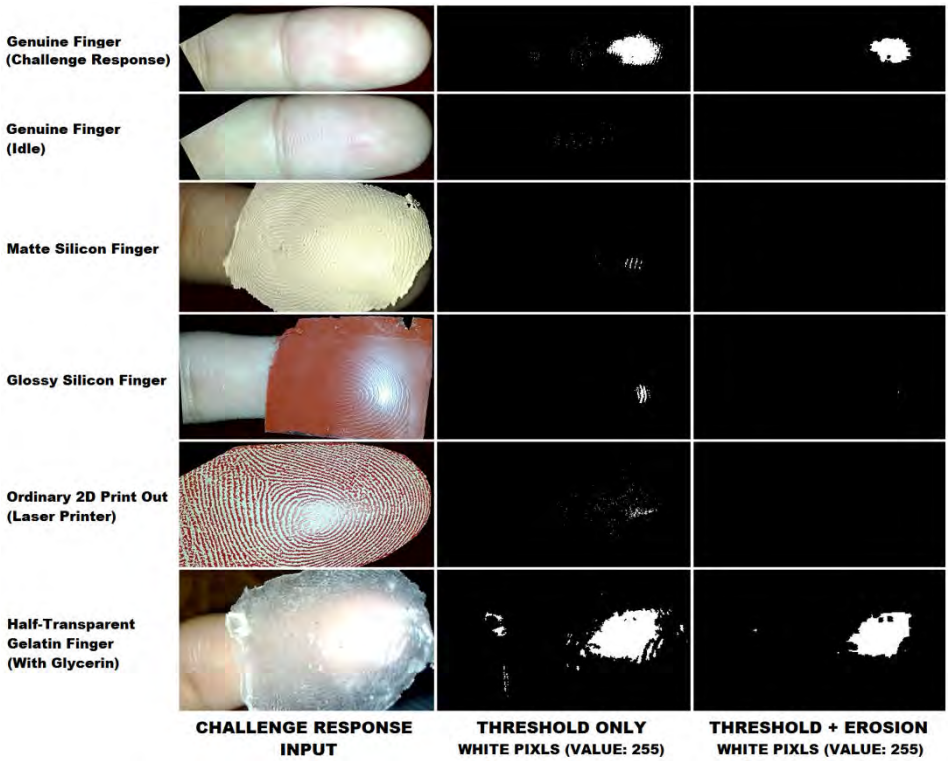

Fig. 3: The genuine finger (with moved fingertip towards the camera) reflects enough light from the LED during the challenge response to pass the challenge response (upper row). The fake fingers reflect (even with shiny and glossy materials) much lower light. However, the fake finger of gelatin with glycerin treatment on its surface (lower row) can also generate a very high reflection like the moved genuine finger.

8.2 Challenge Response

During the challenge response, the position of the finger, the distance of the finger to the camera, the edge density and the light reflection caused from the LED is continuously measured in the video stream:

1. Position of the finger
The measured position of the finger must not differ significantly from the last captured photo of the video stream: The position of the finger is determined with the finger detection algorithm. The position of the left and right boundaries from the last captured ROI and the ROI from the challenge response is checked against the set value (in pixels) for the movement tolerance.

The finger in the photo from the challenge response must also cover the whole area of the quality assurance and must not exceed the image border towards the direction of the fingertip. Otherwise the check fails (even when the measured movement of the finger is lower than the set movement tolerance). The check of the position of the finger keeps the link of the presented finger from authentication mode so it cannot be exchanged by a fake finger. *Figure 4* illustrates a valid example of a position check on a (rotated) fingerphoto: The shadowed area in the center must be covered from the finger but the outer shadowed area must not; the non-shadowed area indicates the allowed movement tolerance.

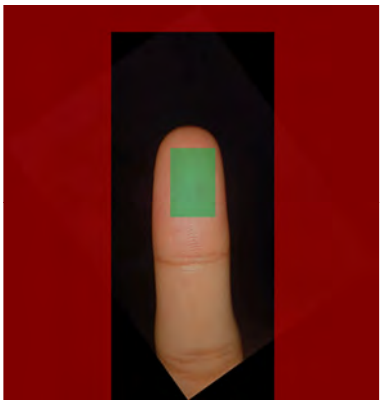

Fig. 4: Valid and invalid areas of the position check

2. Distance of the finger to the camera

The measured width of the finger must not differ significantly from the last captured photo of the video stream: The width of the last captured ROI and the ROI from the challenge response is checked against the set tolerance value. Lowering the distance of the finger to the camera can produce an overexposed image because more light of the LED is captured on the photo on closer distances. The camera compensates too much light incidence by closing the shutter but the correction is delayed. Therefore, fast distance changes of the finger can produce overexposed images before the shutter correction is applied *(see Figure 5).* Those images would achieve higher reflection values and could pass the challenge response falsely. To avoid this issue, distance changes are limited and detected by the change of the fingers width.

Fig. 5: Overexposed fingerphoto

3. Edge density of the finger

The edge density on the ROI of the challenge response is calculated and must not exceed a defined maximum threshold. Print outs from a printer have a typical raster pattern *(see Figure 6).* The raster pattern causes a very high edge density value of 10+. This check detects the usage of a print out during the challenge response.

Fig. 6: Magnified part of a finger print out from a laser printer

4. Light reflection in the inner area of the ROI

The measured light reflection in the core area of the finger *(see next Sub-Section 8.3)* must exceed a threshold: A strong reflection near the fingertip appears due the movement of the fingertip towards the camera and must exceed the threshold for a positive challenge response.

5. Light reflection in the outer area of the ROI
The measured light reflection outside the core area (outer area) of the finger *(see next Sub-Section 8.3)* must not exceed a threshold: Artificial light reflections can be produced from high reflecting materials (in the background) and can be "guided" near the fingertip to achieve a positive challenge response. This measure detects presentation attacks.

For a positive challenge response, all mentioned criteria 1 to 5 must be fulfilled. The challenge response is unresolved and will be continued when all criteria are fulfilled except criterion 4. If one criterion (except 4) is not fulfilled, then the challenge response is aborted with a negative result. In this case the user must restart the authorization process for a new attempt.

8.3 Light Reflection Measurement

The ROIs of the captured frames of the video stream during the challenge response were calculated and converted into gray-scale images. The half width in the center of the ROI and the upper half height of the ROI will be defined as core area and is used to detect the reflections in the central part of the fingerprint for positive authentication of the challenge response. The outer part is the rest of the ROI with all values set to black of the core area and is used to detect reflections at the edges of the finger for negative authentication of the challenge response *(see Figure 7).* Light reflections in this area do not occur from a genuine finger but from spoof attempts.
Only pixels with a maximum value (full white = 255) are kept to detect the light reflections. All other pixels are set to black. An additional morphological operation “erosion” is done to filter small areas of white pixels those are not large enough. The remaining white pixels are count and summed up for the core and outer area of the finger separately and represent the strength of the light reflection for each area. A higher value represents a stronger measured light reflection.

Fig. 7: Definition of the core and outer area on the ROI

9 Evaluation and Results

A biometric database was created with the single photo capture technique and video-based photo capture technique in user tests to evaluate the EER with the minutia extractor and template comparator from Morpho [Morp]. The usability of the presentation attack detection method was tested in a separate user test to determine the genuine acceptance rate. The presentation attack detection rate (PADR) [ISO-2012] was determined with several fake fingers and methods.

9.1 Recognition Rates

Capture Environment and Data Set

Fingerphotos from 37 data subjects were captured with the single fingerphoto capture method. Six to eight photos from the left and the right index finger with the smartphones "Nexus S" and "Galaxy Nexus" from Samsung were captured in two sessions. The resulting test data set consists of 569 unique fingerphotos from the "Nexus S" and 541 photos form the "Galaxy Nexus". 11 data subjects have participated on capturing fingerphotos with the video-based fingerphoto method. Those photos were captured with the "Galaxy Nexus". Six capture sessions have been performed for capturing the left and right index finger (three sessions each). 15 photos are captured on each session. The resulting test data set consists of 990 unique fingerphotos. All captured photos with both capture technologies were only accepted when they have passed the quality assurance. The fingerphotos were captured indoor in a standard office environment. The rooms were well lightened from natural daylight.

Evaluation Procedure

The algorithms from the MorphoLite SDK were not available for the Android operating system [Andr] at the time of writing this paper. Thus, the found and preprocessed ROIs from the input finger photos were stored on the smartphones file system and then transferred to the PC for the minutiae extraction and template creation. The created templates of the video-based photo capture on each session will be consolidated into one template with the algorithms from the MorphoLite SDK. With all templates from the database genuine and imposter comparisons were computed with MorphoLite SDK in order to determine the error rates.

Results

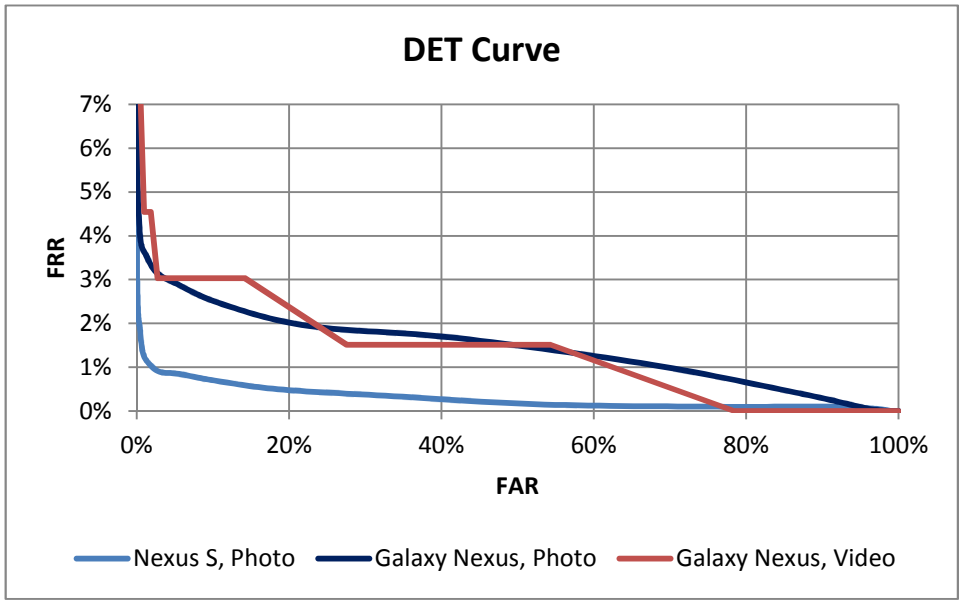

Fig. 8: DET curves of the single photo and video-based capture method on the tested devices

The integration of the minutiae extractor and template comparator from the MorphoLite SDK lead to significant better recognition rates over the results reported in previous work [SC-2012]. The achieved error rates are shown in *Figure 8* in a DET-diagram and in a table in *Figure 9*.

Capture Method and Device	EER from [SC-2012]	EER	FRR (FAR= 0.1%)
Photo, Nexus S	22.3%	1.2%	2.7%
Photo, Galaxy Nexus	19.1%	3.1%	6.7%
Video, Galaxy Nexus	-	3.0%	12.1%

Fig. 9: Achieved error rates

Computation Time / Frame Rate of the Video-based Approach

The measured performance during authentication and enrolment on the "Galaxy Nexus" is about 2.27 frames per second (440ms computing time per frame). 4.55 frames per second (220ms computing time per frame) are achieved during the challenge response. The determined values are the average from the collected values during the user tests.

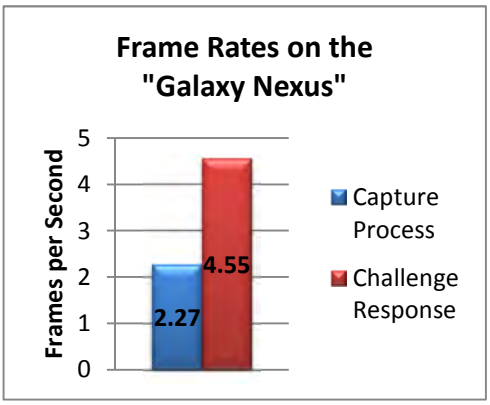

Fig. 10: Processed frames per second of the video-based capture method

9.2 Performance of the Presentation Attack Detection

Genuine Tests

26 subjects have participated on a voluntarily basis to test the challenge response of the application. In order to determine the usability and the false detection rate of the challenge response, each user tries to pass the challenge response with his index finger. Each user has repeated the procedure 10 times after a short instruction and demonstration to the application from the operator. The amount of successful and failed attempts was count. The needed amount of moving the fingertip to the camera was also count in case of a successful challenge response.

Fig. 11: Recognition Rate of the challenge response

Results

201 of 260 performed challenge responses were successful. The result is a recognition rate of 77.3% *(see Figure 11)*. 80 of them were recognized at the first time as the finger was moved to the camera. In 69 cases the fingertip movement must repeated twice to the camera in order to achieve a successful challenge response. In 44 cases three attempts was needed for a successful challenge response. Four or more attempts were needed in 8 cases. Subjects that have held the smartphone in a brighter area e.g. near a window, had more difficulties to pass the challenge response. Light conditions decrease

the effect of the light reflection of the LED and make it harder to pass the challenge response.

Artefact Detection Tests

The following fake tests are done:

- Several fake 2D print outs with original unprocessed and binarized fingers in different sizes printed with a laser printer on ordinary paper *(see Figure 12).*
- Several fake fingers of gummy, silicon and gelatin those differ in color and shape *(see Figure 13).*
- A replay attack with a video captured and injected with another "Galaxy Nexus" smartphone *(see Figure 14).*

Fig. 12: Variety of tested fake 2D print outs

Fig. 13: Variety of tested fake fingers of gummy, silicon and gelatin

Fig. 14: Replay attack with a video on another smartphone

Results

Many repeated fake attempts with the above mentioned fake attacks were performed. The determined true PADR is 0.83 based on the tests of the six different attack presentation characteristics: 2D print out, fake finger of gummy, silicon and gelatin, fake finger with post treatment and replay attack. The results give first impressions about the potential of the developed anti spoofing technique. Extended tests are needed to make more meaningful statements about the spoofing resistance.

2D print outs

The fakes possess a higher edge density value due the raster pattern effect *(see Figure 6)* and are normally detected by the edge density check. However, the fake can pass this check when the fake is not properly in focus. But such a fake is not able to pass the reflection check because the material and the raster pattern effect (only the pigments of the toner are reflected) do not provide such high reflection strength as a genuine finger.

Fake fingers of gummy, silicon and gelatin

None of the tested (unprocessed) fake fingers was able to produce the necessary light reflections to pass the challenge response. However, a half-transparent fake finger of gelatin with a special treatment of the surface with glycerin can simulate a similar reflection behavior of a real finger and was able to pass the challenge response.

Video replay attack

The reflection strength from the shown finger on the display in the video is far lower

than a natural reflection. The surface of the (glossy) screen itself reflects much more light and makes a proper capture respectively an injection almost impossible resulting in a failure of the challenge response due various finger detection errors like too small ROI or too high finger movement. After all, the measured reflection of the screen is still lower and do not pass the challenge response.

10 Conclusions and Future Work

A new approach to capture fingerphotos over a video stream with a smartphone camera has been implemented and evaluated. An EER of about 3% was achieved. The existing prototype has been improved in aspects of recognition rate, usability and anti-spoofing resistance. However, a smartphone with at least a fast dual-core processor is needed to achieve a usable frame rate. Otherwise the CPU is not able to process the frames in time and a lot frames must be discarded due the lack of available CPU resources which results in a decreased capture rate and decreased anti-spoofing detection rate. The developed anti-spoofing technique can detect spoof attempts with fake print outs and fake fingers of gummy, silicon and gelatin as well video replay attacks. However, advanced techniques with special treatments of the surfaces of finger fakes can simulate similar light reflections of a real finger those cannot detected reliably with the current implementation of this technique.
Further development can be the modification of the finger-detection algorithm to detect multiple fingers per frame in the video stream. This will decrease the effective needed time per capture further and enables a convenient way to capture multiple fingerprints from different fingers from one subject.

Acknowledgement

This work was partly sponsored by Safran Morpho group. The authors also like to thank the volunteers that supported the technology and usability testing in this work.

References

[DM-2011] Derawi, Mohammad Omar; Yang, Bian; Busch, Christoph: *Fingerprint Recognition with Embedded Cameras on Mobile Phones*. In: MobiSec 2011, Denmark, 2011

[ISO-2012] ISO/IEC JTC1 SC37 Biometrics: *ISO/IEC 30107. Information Technology - Biometrics - Presentation attack detection.* ISO, 2012

[LD-2008] Lee, Dongjae; Choi, Kyoungtaek; Choi, Heeseung; Kim, Jaihie: *Recognizable-Image Selection for Fingerprint Recognition with a Mobile-Device Camera.* In: IEEE Transactions on Systems, Man and Cybernetics, 2008, Vol. 38, p. 233-243

[Morp] Morpho: *Corporation Homepage*. http://www.morpho.com
Last checked: 2013-07-03

[OSCV] Open Source Computer Vision: *OpenCV Homepage*. http://opencv.org
Last checked: 2013-07-03

[SC-2012] Stein, Chris; Nickel, Claudia; Busch, Christoph: *Fingerphoto Recognition with Smartphone Cameras.* In: BIOSIG 2012 - Proceedings, Germany, 2012, p. 87-98

[WH-2010] Witte, Heiko; Nickel, Claudia: *Modular Biometric Authentication Service System (MBASSy).* In: BIOSIG 2010 - Proceedings, Germany, 2010, p. 115-120