REAL TIME EYE BLINKING FOR PASSWORD AUTHENTICATION

Mahesh TR[1], M.Sai Ram[2], N.Sai Ram[3], Allu Gowtham[4], T.V. Narayana Swamy[5]

[1]Professor, Dept. of computer science Engineering, SET-Jain University, Benguluru- Karnataka, India
[2-5]Student , Dept. of computer science Engineering, SET-Jain University, Benguluru-Karnataka, India

---

Abstract— Personal identification numbers are widely used for user authentication and security. Password verification using PINs requires users to enter a physical PIN, which can be vulnerable to password breakage or hacking. via shoulder surfing or thermal tracking. PIN authentication with hands-off eye blinks PIN entry techniques, on the other hand, leaves no physical footprints behind and therefore offer a more secure password entry option. Eye blinks based authentication refers to finding the eye blinks across sequential image frames, and generating the PIN. This project presents a real-time application we combine eye blink-based PIN entry, and face detection and OTP(One Time Password) to avoid shoulder surfing and thermal tracking attacks.

1. INTRODUCTION

One of the security requirements for general terminal authentication systems is to be easy, fast and secure as people face authentication mechanisms every day and must authenticate themselves using conventional knowledge-based approaches like passwords. But these techniques are not safe because they are viewed by malicious observers who use surveillance techniques such as shoulder-surfing (observation user while typing the password through the keyboard) to capture user authentication data. Also there are security problems due to poor interactions between systems and users. As a result, the researchers proposed a three layered security framework to secure PIN numbers, where users can enter the password by blinking the eye at the suitable symbols in the appropriate order and thus the user is invulnerable to shoulder surfing. Eye blinking is a natural interaction method and security systems based on eye blink tracking provide a promising solution to the system security and usability. The aim of this paper is to review techniques or solutions to dealing with eye blink in security systems.

The use of personal identification numbers (PINs) is a common user authentication method for many applications, such as money management in automatic teller machines (ATMs), approving electronic transactions, unlocking personal devices, and opening doors. Authentication is always a challenge even when using PIN authentication, such as in financial systems and gateway management. According to European ATM Security, fraud attacks on ATMs increased by 26% in 2016 compared to that of 2015. The fact that an authorized user must enter the code in open or public places make PIN entry vulnerable to password attacks, such as shoulder surfing as well as thermal tracking.

2. ALGORITHM SPECIFICATION

HAAR CASCADE CLASSIFIER:

The acquisition of an object using the Haar-based platform is an effective method of object discovery proposed by Paul Viola and Michael Jones in their paper, "Recent Acquisitions Using Enlarged Cascade. It is a machine learning technique where Cascade's work is trained from many positive and negative images. After that it is used to find objects in other images .Initially, the algorithm requires many good images (face images) and negative images (images without faces) to train the student. After that we need to extract features from it. In this case, using the Haar features shown in the image below.

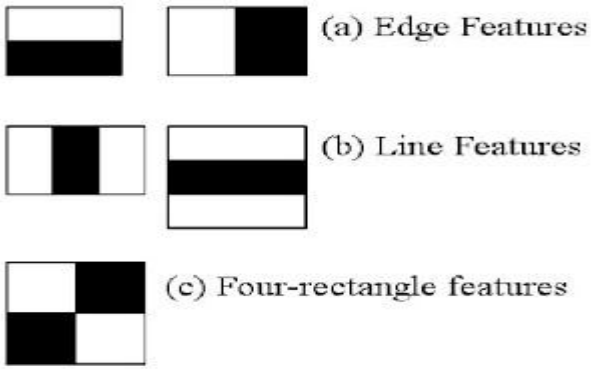

FIG: 2.1 Different Features

Now, all possible sizes and locations of each kernel are used to calculate multiple features. To calculate each element, we need to find the sum of pixels under the white and black squares. To solve this, they introduced an important image. Even though your image is large, it limits the pixel count provided to functionality that involves just four pixels. But among all the features we have listed, many of them do not work. The first feature selected seems to focus on the properties of the eye region that is often darker than the nose and tire region.But among all these features we calculated, most of them are irrelevant. For example, consider the image below. The top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applied to cheeks or any other place is irrelevant. So

how do we select the best features out of 160000+ features? It is achieved by Adaboost.

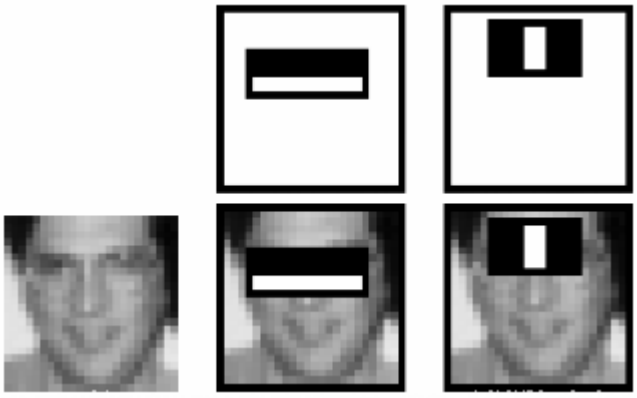

FIG: 2.2 Focusing the Features

In this case, we apply each element to all training images. At each stage, you get a really good edge that will distinguish the face from good and bad. We select the features with a small error rate, which means they are the features that distinguish the faces from the non-face images. (The process is not as simple as this. Each image is given the same weight at the beginning. After each split, the illegals of the statue increase. The process continues until the accuracy or error rate is reached or the number of features is reached).The final classifier is a weighted sum of these weak learners. It is called weak because it alone will not distinguish the image, but in association with others it creates a strong bonding phase.

The paper states that even 200 features provide 95% accuracy. (Consider a reduction from 160000 features to 6000 features. The authors have a good solution for that. For an image r, most of the image is surface area. So it's a good idea to have a simple way to check that a window is not a surface. If not, get rid of it by firing one, and never do it again. Instead, focus on regions where there may be faces. This way, we spend a lot of time looking at potential regions of the face. They presented this concept in the Cascade of Classifiers.

Instead of using all 6000 objects in a window, the features are organized into different classes of learners and worked individually. (Generally the first few paragraphs will contain very few features).If a window fails in the first section, discard it. If it passes, use the second phase of the symptoms and continue the process. The highest window of all categories is the face area.The authors' detector had 6000+ features with 38 stages with 1, 10, 25, 25 and 50 features in the first five stages. (The two features in the above image are actually obtained as the best two features from Adaboost). According to the authors, on average 10 features out of 6000+ are evaluated per sub-window. So this is a simple intuitive explanation of how Viola-Jones face detection works. Read the paper for more details or check out the references in the Additional Resources section.

CNN Model:

This step is the most important part of the entire process as we design the CNN through which we will pass our features to train the model and eventually test it using the test features. We have used a combination of several different functions to construct CNN which we will discuss one by one.

1. **Sequential()** - A sequential model is just a linear stack of layers which is putting layers on top of each other as we progress from the input layer to the output layer.
2. **model.add(Conv2D())** - This is a 2D Convolutional layer which performs the convolution operation as described at the beginning of this post. To quote Keras Documentation "This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs." Here we are using a 3x3 kernel size and Rectified Linear Unit (ReLU) as our activation function.
3. **model.add(BatchNormalization())** - It performs the batch normalization operation on inputs to the next layer so that we have our inputs in a specified scale say 0 to 1 instead of being scattered all over the place.
4. **model.add(MaxPooling2D())** - This function performs the pooling operation on the data as explained at the beginning of the post. We are taking a pooling window of 2x2 with 2x2 strides in this model
5. **model.add(Dropout())** - As explained above Dropout is a technique where randomly selected neurons are ignored during the training. They are "dropped out" randomly. This reduces overfitting.
6. **model.add(Flatten())** - This just flattens the input from ND to 1D and does not affect the batch size.
7. **model.add(Dense())** - According to Keras Documentation, Dense implements the operation: output = activation(dot(input, kernel)where activation is the element-wise activation function passed as the activation argument, kernel is a weights matrix created by the layer. In simple words, it is the final nail in the coffin which uses the features learned using the layers and maps it to the label. During testing, this layer is responsible for creating the final label for the image being processed.

2.1 MODULES

2.1.1 FACE RECOGNITION USING CNN

Let's look first on how human do recognize the faces. Face perceptions are very complex as the recognition of facial expressions involves extensive and diverse areas in the brain. Brain imaging studies typically show a great deal of activity in an area of the temporal lobe known as

the fusiform gyrus, an area also known to cause prosopagnosia when damaged (particularly when damage occurs on both sides). People learn to recognize faces from birth and nearly at the age of four months can clearly distinguish one person from another.

The main thing that a person pays attention to is the eyes, cheekbones, nose, mouth, and eyebrows, as well as the texture and color of the skin. At the same time, our brain processes the face as a whole and is able to identify a person even by half of the face. The brain compares the resulting image with the internal measurement pattern and receives the positional difference.

First, the face recognition system needs to find the face in the image and highlight this area. For this, the software can use a variety of algorithms: for example, determining the similarity of proportions and skin color, the selection of contours in the image and their comparison with the contours of faces, the selection of symmetries using neural networks. The most effective is the **Viola-Jones method**, which can be used in real time. With it, the system recognizes faces even when rotated 30 degrees.

The method is based on the signs of Haar, which are a set of black and white rectangular masks of different shapes. The masks are superimposed on different parts of the image, and the algorithm adds the brightness of all the pixels of the image that are under the black and white parts of the mask and then calculates the difference between these values. Next, the system compares the results with the accumulated data and, having determined the face in the image, continues to track it to select the optimum angle and image quality. For this purpose, motion vector prediction algorithms or correlation algorithms are used.

Having chosen the most successful pictures, the system proceeds to face recognition and its comparison with the existing base. It works according to the same principles as the artist paints portraits — the program finds the reference points on the person's face that make up the individual features. As a rule, the program allocates about 100 such points.

The most important measurements for face recognition programs are the distance between the eyes, the width of the nostrils, the length of the nose, the height and shape of the cheekbones, the width of the chin, the height of the forehead and other parameters. After that, the obtained data are compared with those available in the database, and, if the parameters coincide, the person is identified.

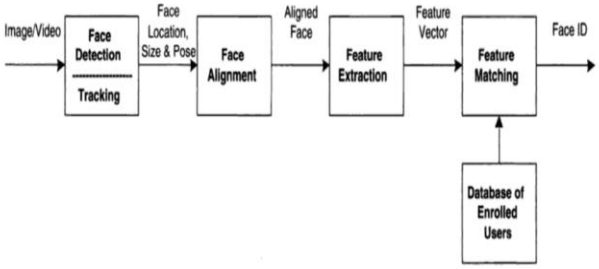

Fig. 1.2. Face recognition processing flow.

FIG:2.1.1 Face Recognition Processing Flow

2.1.2EYE BLINK PASSWORD GENERATION

This is the second layer authentication in our project we are displaying the digital keyboard on the screen. One cursor will keep moving in the digital numeric keyboard when user blinks eye system will generate the PIN number from collect the sequence of numbers which user blinks eye. We are using the web camera to monitor eye moments. We are using OpenCV to detects eye blink.

We will build upon this knowledge and develop a computer visualization app that can detect and calculate video stream crashes using face bookmarks and OpenCV.

To build our more accurate detector, we will be using a metric called eye factor ratio (EAR).
Unlike traditional computer animation techniques that involve some combination of:

1. Eye localization.
2. Thresholding to find the whites of the eyes.
3. Determining if the "white" region of the eyes disappears for a period of time (indicating a blink).
4. The eye aspect ratio is instead a much more elegant solution that involves a very simple calculation based on the ratio of distances between facial landmarks of the eyes.

2.1.3OTP GENERATION AND VERIFICATION

This is the third level of security we are using the random numbers to generate the otp and send to users email/ mobile number using the otp we can secure our accounts.

3. IMPLEMENTATION

Localization of Face: Since the Face is symmetric, we use a symmetry-based approach. We found that it is sufficient to use a sample version, which is a gray scale. The symmetry-value is calculated and then calculated across the pixel columns in the reduced image. If the image is represented as I(x, y) then the symmetry value for a pixel-column is given by $S(x) = \sum\sum$ [abs I ((x, y-w)-(x, y+w))].

S(x) is computed for X € [k, size-k] where k is the maximum distance from the pixel-column that symmetry is measured, and x size is the width of the image. The x corresponding to the lowest value of S(x) is the center of the face.

Tracking of the eyes: We track the eye by looking for the darkest pixel in the predicted region. In order to recover from tracking errors, we make sure that none of the geometrical constraints are violated. If they are, we re-localize the eyes in the next frame. To find the best match for the eye template, we initially center it at the darkest pixel, and then perform a gradient descent in order to find a local minimum.

We are going to propose the three layer security scheme to avoid the shoulder surfing and thermal tracking attacks. Our system contains the three layers which are 1.Face reorganization, 2. Eye- blink verification, and 3. OTP by combining all this layers we are going to implement our secure framework to avoid shoulder surfing and thermal tracking attacks. In our frame works there is no physical entry of password so we are completely avoiding the shoulder surfing and thermal tracking attacks. For the first layer security we are using Deep Learning algorithm., for the second layer we are using OpenCV.

For the background positions of the eye, we get ° ng based on the differences between the pictures respectively.

To separate them from the headaches, we emphasize the head wiping process. Trace eye, using the template of' Between the Eyes, 'is renewable each framework, instead of the eyes. The eyes are said to be based on the current position 'Between-the-Eyes' and their geometrical relationship to the previous position.

4. SYSTEM ARCHITECTURE

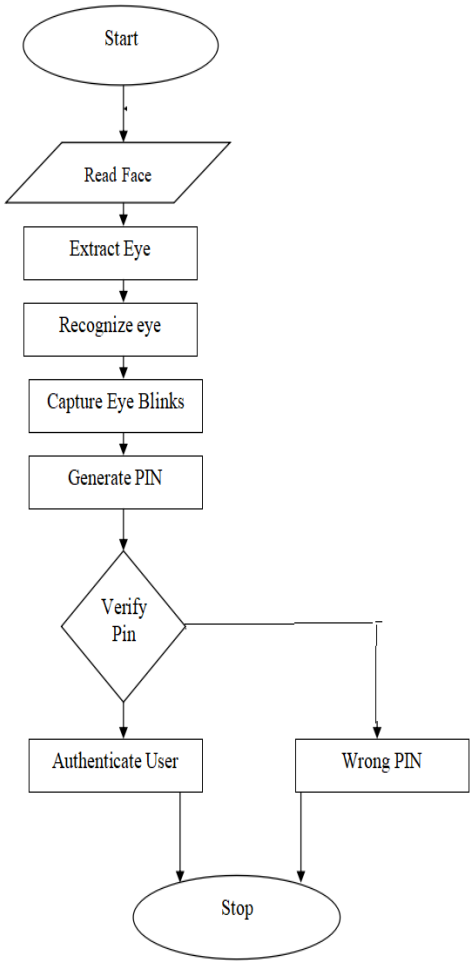

FIG:4.1 System Architecture

4. DATAFLOW DIAGRAM

The DFD is also referred to as bubble chart.It is an easy graphical formalism that can be used to represent a system,varied process distributed on this data,and therefore the output data is generated by this system

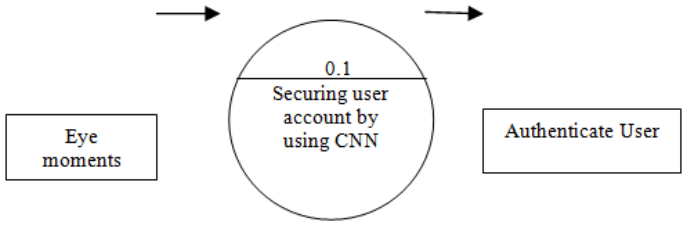

FIG:5.1

We are using users eye moments as input. System will use the conventional neural network to secure user account information from shoulder surfing attacks.

Level:1

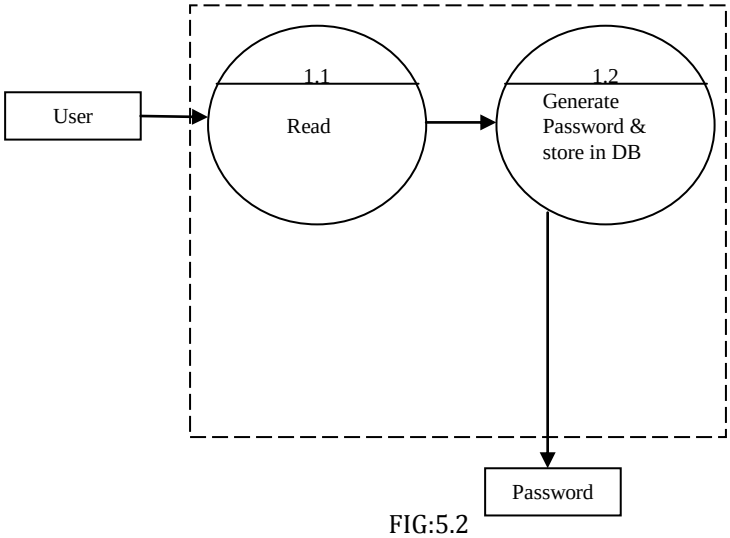

FIG:5.2

Level:2

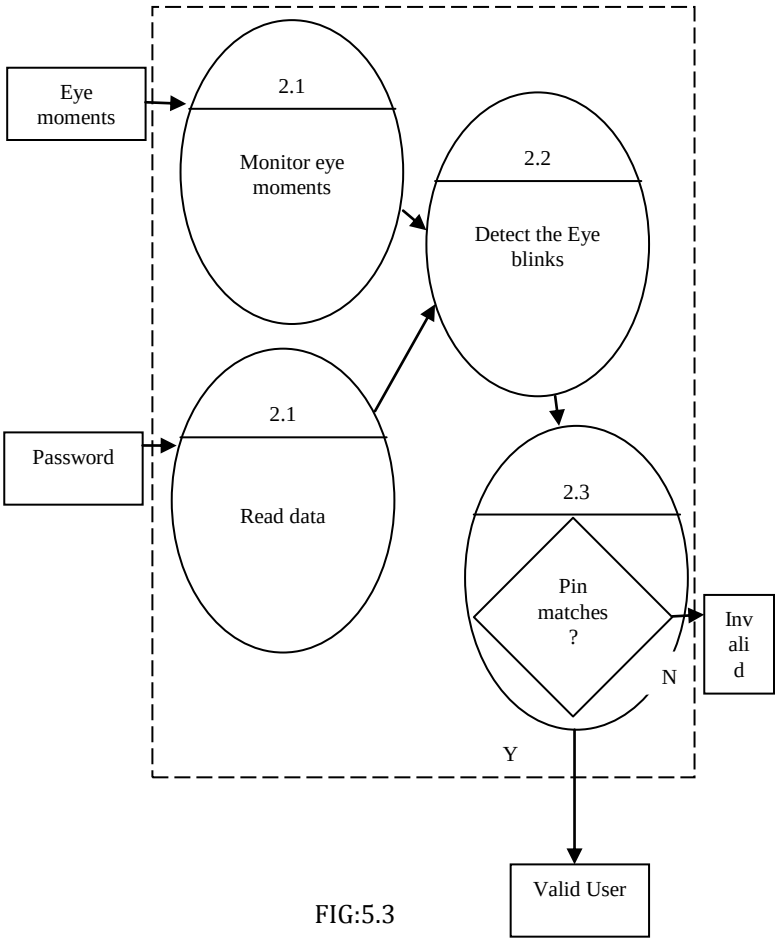

FIG:5.3

5. CONCLUSION

A smart-camera based eye-blinking system has been incorporated into a new application for eyelid blink-based PIN identification. The system has been successfully tested with a nine-digit keypad, and can be extended to character and digit combination password entry.. The stability of the user's eye blink will affect the accuracy of the detected pins, and must be accounted for. Currently, the PIN identification is accomplished after real-time eye-blinks and eye center computations and recording are completed.

6. REFERENCES

[1] R. Revathy and R. Bama, "Advanced Safe PIN-Entry Against Human Shoulder-Surfing," IOSR Journal of Computer Engineering, vol 17, issue 4, ver. II, pp. 9-15, July-Aug. 2015. (Available: http://www.iosrjournals.org/iosr-jce/papers/Vol17-issue4/Version2/B017420915.pdf)

[2] J. Weaver, K. Mock and B. Hoanca, "Gaze-Based Password Authentication through Automatic Clustering of Gaze Points," Proc. 2011 IEEE Conf. on Systems, Man and Cybernetics, Oct. 2011. (DOI: 10.1109/ICSMC.2011.6084072)

[3] "ATM Fraud, ATM Black Box Attacks Spread Across Europe", European ATM Security Team (E.A.S.T.), online, posted 11 April 2017. (Available: https://www.european-atm-security.eu/tag/atmfraud/)

[4] K. Mowery, S. Meiklejohn and S. Savage, "Heat of the Moment: Characterizing the Efficacy of Thermal Camera-Based Attacks," WOOT '11, pp. 1-8, August 2011. (Available: https://cseweb.ucsd.edu/~kmowery/papers/thermal.pdf)

[5] M. Mehrübeoglu, H. T. Bui and L. McLauchlan, "Real-time iris tracking with a smart camera," Proc. SPIE 7871, 787104, 2011. (DOI:10.1117/12.872668)

[6] M. Mehrubeoglu, L. M. Pham, H. T. Le, M. Ramchander, and D. Ryu, "Real-time eye tracking using a smart camera," Proc. 2011 IEEE Applied Imagery Pattern Recognition Workshop (AIPR '11), pp. 1-7, 2011. (DOI: 10.1109/AIPR.2011.6176373)

[7] M. Mehrubeoglu, E. Ortlieb, L. McLauchlan, L. M. Pham, "Capturing reading patterns through a real-time smart camera iris tracking system," Proc. SPIE, vol. 8437, id. 843705, 2012. (DOI: 10.1117/12.922875)

[8] Smart Cameras for Embedded Machine Vision, (product information) National Instruments (Available: http://www.ni.com/pdf/products/us/cat_ni_1742.pdf)